



## **Конфигурации Doc Ext**

Приложение для IBM Lotus Notes и Domino®.

Версия 2.0.2

**Руководство разработчика и администратора**

## Оглавление

1	Описание приложения .....	3
2	Документы конфигурации.....	4
2.1	Описание.....	4
2.2	Тип DocExt .....	4
2.2.1	Тип DocExt - Пункты Doc.....	5
2.2.2	Тип DocExt - Пункты DocForDoc.....	7
2.2.3	Тип DocExt - Пункты DocUpDoc .....	9
2.3	Тип DocToDoc .....	11
2.4	Тип Project .....	14
3	Библиотека DocExtLib .....	16
3.1	Класс DocExt .....	16
3.2	Публичные функции .....	17
3.2.1	Функция DEdocDescr.....	18
3.2.2	Функция DEdocID.....	18
3.2.3	Функция DEdocLocked .....	18
3.2.4	Функция DEgetApp.....	19
3.2.5	Функция DEgetAppName .....	19
3.2.6	Функция DEgetC4D.....	20
3.2.7	Функция DEgetD4D .....	20
3.2.8	Функция DEgetPrjName .....	21
3.2.9	Функция DEgetSrcC4D.....	21
3.2.10	Функция DEgetSrcD4D.....	22
3.2.11	Функция DEnotGetC4D.....	23
3.2.12	Функция DEopenD4D .....	24
3.2.13	Функция DEpickC4D .....	24
3.2.14	Функция DEpickD4D .....	25
3.2.15	Функция DEpickDocs .....	26
3.2.16	Функция DEpickSrcC4D.....	27
3.2.17	Функция DEpickSrcD4D .....	28
3.2.18	Функция DEprCopyDelay .....	29
3.2.19	Функция DEprD2D .....	29
3.2.20	Функция DEprD2Ddelay.....	30
3.2.21	Функция DEprDelay.....	31
3.2.22	Функция DEprDelete .....	31
3.2.23	Функция DEprMove.....	32
3.2.24	Функция DEpromptApp .....	32
4	Настройки .....	33
4.1	Начало использования приложения .....	33
4.2	Журнал обновления документов .....	33
4.3	Отложенное обновление документов .....	35
4.4	Переменные библиотеки кода.....	36

## 1 Описание приложения

Приложение Конфигурации Doc Ext ( DEC для краткости ) позволяет централизованно управлять данными документов в приложениях IBM Lotus Notes® с минимальным написанием кода в них. Опишите в конфигурациях приложения DEC документы ваших приложений, а также как один документ получает другие и как обновляет их. Кроме того, приложение предоставляет расширенные возможности для журнализации всех изменений документов в ваших приложениях.

Приложение DEC содержит единственную библиотеку Lotus Script для вставки её в ваши приложения и определяемые администратором документы конфигурации. Дополнительно приложение содержит несколько элементов дизайна для развертывания журнализации изменений документов и отложенного обновления документов с помощью сервера.

### **Совместимость**

Версия приложения 2.0.2 была тщательно протестирована и наблюдалась ее корректная работа в IBM Lotus Notes и Domino® версий 6.x.x - 9.x.x с операционными системами MS Windows® версий XP, Vista, 7, 8, 8.1, 10 в любых сочетаниях.

### **Разработчик**

Разработчиком и правообладателем приложения является Вадим Матвиенко. Для любых вопросов, связанных с работой приложения, используйте адрес электронной почты: [support@apps4notes.org](mailto:support@apps4notes.org). Сайт приложения: <http://www.apps4notes.org/ru>.

### **Лицензия**

Конфигурации Doc Ext является свободным программным обеспечением и лицензировано согласно Лицензии Apache, Версия 2.0 ( "Лицензия" ); вы можете использовать этот файл только в соответствии с Лицензией. Вы можете найти копию Лицензии по адресу <http://www.apache.org/licenses/LICENSE-2.0.txt>. За исключением случаев, когда это регламентировано существующим законодательством или, если это не оговорено в письменном соглашении, программное обеспечение, распространяемое на условиях данной Лицензии, предоставляется "КАК ЕСТЬ" и любые явные или неявные ГАРАНТИИ ОТВЕРГАЮТСЯ. Информацию об основных правах и ограничениях, применяемых к определенному языку согласно Лицензии, вы можете найти в данной Лицензии.

### **Что нового в версии 2.0.2 ?**

- Расширена проверка синтаксиса выражений на Языке Формул в конфигурациях;
- Публичная функция DEOpenD4D стала возвращать результат ( True или False ), а также может быть вызвана в режиме Silent;
- Изменения документа теперь продолжают регистрироваться в нем даже при превышении размера поля для журнализации 32КБ;
- Добавлена глобальная переменная DE\_PRINT для отключения вывода на экран любых сообщений при работе приложения;
- Добавлено описание аргументов во всех функциях библиотеки кода DocExtLib;
- Исправлено несколько небольших ошибок.

## 2 Документы конфигурации

### 2.1 Описание

Документы конфигурации хранят все конфигурационные данные о документах в ваших приложениях. Существует три типа конфигурационных документов: DocExt ( включает пункты Doc, DocForDoc и DocUpDoc ), DocToDoc и Project.

Любая конфигурации состоит из его пунктов. Каждый пункт пишется на новой строке. Каждый пункт в зависимости от типа конфигурации может состоять либо из наименования и следующих за ним тегов, либо только из тегов. Тег определяет либо название чего-либо, либо действие над чем-либо. Тег может быть написан сам по себе или со скобками [ ], { }, < >. Внутри скобок [ ] необходимо вписывать текст, чтобы задать значение тега. Внутри скобок { } необходимо вписывать условие или выражение на Языке Формул, чтобы вычислять значение тега или его применение. Впишите тег со скобками < > с названием другого пункта конфигурации внутри, чтобы использовать значение одноименного тега из указанного пункта. Какие скобки могут быть использованы с тем или иным тегом читайте в описании тега в настоящей справке.

Если скобки тега также используются в значении этого тега, то необходимо использовать полный формат написания тега с добавлением наименования тега после скобок: TAG{ ... SUBTAG{ ... } ... }TAG.

Чтобы проверить синтаксис всех выражений на Языке Формул, написанных во всех тегах со скобками { } всех пунктов конфигурации, нажмите "Проверить весь синтаксис" вверху документа конфигурации.

Чтобы отключить какой-либо пункт конфигурации, напишите ключевое слово REM в начале этого пункта.

### 2.2 Тип DocExt

#### Описание

DocExt это основной тип документов конфигурации в приложении DEC. DocExt описывает один тип ( форму ) документов в каком-либо приложении. Он состоит из трех частей: Doc, DocForDoc, DocUpDoc. Каждая часть может быть включена / выключена независимо.

#### Добавление

Чтобы создать конфигурацию DocExt выполните:

- Откройте представление **Конфигурации \ DocExt** и в его меню нажмите **Создать конфигурацию DocExt**;
- На открывшейся форме в поле **Наименование** впишите наименование конфигурации без пробелов. Допускается два наименования, первое - краткое для удобства, второе ( на новой строке ) - наименование совпадающее с формой документов, описываемых в данной конфигурации. Например: cntr и Contract. Оба наименования должны быть уникальны среди всех конфигураций DocExt. Вы можете не писать краткое наименование, но имя формы обязательно;

- Справа определите, какие части конфигурации включены: **Doc, DocForDoc, DocUpDoc**;
- В поле **Описание** впишите подробный текст о документах с указанной формой. Например: Договоры предприятия. Описание может показываться пользователю при выполнении кода.
- В полях **Версия** установите номер и дату версии конфигурации. Это справочная информация, которая не будет использована где-либо.

### 2.2.1 Тип DocExt - Пункты Doc

#### Описание

Пункты конфигурации Doc являются частью конфигурации DocExt и описывают тип ( форму ) документов сам по себе. Для документа в целом пункты конфигурации определяют его проверку на уникальность, возможность быть измененным или удаленным ( блокировка ), журнализацию изменений и другое. Для поля документа пункт конфигурации проверяет его на заполнение и допустимое значение.

Пункты конфигурации Doc необходимо вводить на закладке Doc конфигурации DocExt. Используйте новую строку для каждого пункта. Первые пункты конфигурируют документ в целом, и каждый последующий пункт конфигурирует поле документа.

К пунктам Doc можно обращаться каждый раз перед сохранением документа.

#### Формат пункта

Используйте первые пункты ( строки ) конфигурации Doc для настройки документа в целом.

Формат пункта, конфигурирующего документ:

**TYPE[ наименование ] NAME{ формула } LOG\_FIELD[ наименование ] LOG\_APP[ наименование ]  
LOG\_MAIL[ подтеги ] LOCK{ условие } LOCK\_D4D[ наименование ] UNIQ\_D4D[ наименование ]**

Описание формата:

**TYPE[ наименование ]** - определяет наименование типа документа, которое будет отображаться перед наименованием документа в каком-либо сообщении при выполнении кода. Если Вы не укажете этот тег, то наименование формы документа будет использовано;

**NAME{ формула }** - вычисляет наименование документа для какого-либо сообщения при выполнении кода. Если не укажете этот тег, то UNID документа будет использован;

**LOG\_FIELD[ наименование ]** - определяет поля, в которое будет записываться журнал изменений документа. Если Вы не укажете этот тег, то наименование поля по умолчанию будет использовано ( устанавливается константой DE\_LOG\_FIELD в Библиотеке ). Если Вы не хотите журнализировать изменения в поле, впишите тег без наименования;

**LOG\_APP[ наименование ]** - определяет приложение ( в каком-либо проекте ), в которое будет сохраняться журнал изменений документа. Проекта и их приложения определяются в конфигурациях типа Project. Если не укажете этот тег, то наименование приложения по умолчанию будет использовано ( устанавливается константой DE\_LOG\_APP в Библиотеке ). Если Вы не хотите журнализировать изменения в приложение, впишите тег без наименования;

**LOG\_MAIL[ подтеги ]** - определяет отправку уведомления об изменении документа. Тег состоит из подтегов внутри:

**LOG{ условие }** - определяет когда уведомление будет отправлено. Условие проверяется на документе журнала ( не на измененном документе ), имеющим единственное поле Message с содержанием изменений;

**DOC{ условие }** - определяет когда уведомление будет отправлено. Условие проверяется на измененном документе;

**TO[ адрес ]** - определяет получателя уведомления. Вы также можете написать TO{ формула }, чтобы вычислять ( на измененном документе ) один или несколько адресов электронной почты;

**SUBJ[ текст ]** - определяет тему сообщения. Тема будет добавлена после наименования измененного документа, вычисленного с помощью тега NAME{ формула }. Напишите тег SUBJ{ формула }, чтобы вычислять тему на измененном документе.

**LOCK{ условие }** - определяет блокировку документа, что запрещает его редактирование и удаление. Результат условия @True (1) или какой-либо текст для документа блокирует его. Текст будет добавлен в окно сообщения. Вы также можете использовать этот тег в пункте конфигурирования поля;

**LOCK\_D4D[ наименование ]** - работает как LOCK{ условие }, но документ блокируется, если будут получены документы по пункту конфигурации DocForDoc. Вы также можете использовать этот тег в пункте конфигурирования поля;

**UNIQ\_D4D[ наименование ]** - определяет, что документ не уникален, если получены документы по пункту конфигурации DocForDoc. Рекомендуется использовать этот тег в пункте конфигурирования поля.

Пункт конфигурирования поля должны размещаться под пунктами конфигурирования документа.

Формат пункта конфигурирования поля:

**наименование [ описание ] LOCK{ условие } LOCK\_D4D[ наименование ] FILL FILL{ условие } FAIL{ условие } FAIL\_D4D[ наименование ] VLD{ условие } VLD\_D4D[ наименование ] UNIQ\_D4D[ наименование ] UNIQ UNIQ{ условие }**

Описание формата:

**наименование** - имя поля, которое конфигурируется;

**[ описание ]** - описание поля. Будет использовано в сообщениях выполнения кода и журнализации изменений;

**FILL** - определяет, что поле должно быть заполнено;

**FILL{ условие }** - определяет когда поле должно быть заполнено;

**FAIL{ условие }** - определяет когда значение поля недопустимо;

**FAIL\_D4D[ наименование ]** - означает, что значение поля недопустимо, если получены документы по пункту конфигурации DocForDoc;

**VLD{ условие }** - определяет когда значение поля допустимо;

**VLD\_D4D[ наименование ]** - означает, что значение поля допустимо только если получены документы по пункту конфигурации DocForDoc;

**LOCK{ условие }** - определяет когда документ заблокирован, т.е. не может быть изменен и удален;

**LOCK\_D4D[ наименование ]** - работает как LOCK{ условие }, но в случае, если получены документы по пункту конфигурации DocForDoc;

**UNIQ\_D4D[ наименование ]** - означает, что документ не уникален, если получены документы по пункту конфигурации DocForDoc. Используйте этот тег для первого поля, участвующего в теге KEY пункта конфигурации DocForDoc;

**UNIQ** - напишите этот тег, если поле принимает участие во всех пунктах конфигураций DocForDoc, указанных в теге UNIQ\_D4D других пунктов конфигурации выше;

**UNIQ{ условие }** - работает как UNIQ, но только если результат условия @True (1).

Для всех формул и условий в тегах конфигурации ожидаются результаты: @True (1), @False (2) или непустой текст. Если получен ожидаемый результат, то окно сообщения с описанием и значением поля будет показано. Если результатом является непустой текст, тогда он будет добавлен в окно сообщения.

Теги LOCK\_D4D, FAIL\_D4D, VLD\_D4D, UNIQ\_D4D могут быть использованы более одного раза в пункте конфигурации. Кроме того, Вы можете использовать расширенный формат вызова пункта конфигурации DocForDoc: ...[ SRC{ условие } наименование TRG{ условие } ], где условие в теге SRC будет применено к исходному документу ( который проверятся ) и только если результатом условия будет @True (1), то получение документов по пункту конфигурации DocForDoc с указанным наименованием будет выполнено. Впишите условие в тег TRG, чтобы проверять его для полученных документов - только документы с результатом проверки @True (1) будут оставлены в качестве результата.

## 2.2.2 [Тип DocExt - Пункты DocForDoc](#)

### [Описание](#)

Пункты конфигурации DocForDoc являются частью конфигурации DocExt и описывают, как один документ получает другие. Пункты конфигурации предоставляют множество возможностей для простого получения и выбора документов из представлений в Ваших приложениях.

Пункты конфигурации DocForDoc должны вноситься на закладке DocForDoc конфигурации DocExt. Используйте новую строку для каждого пункта. Первый пункт конфигурирует значения по умолчанию и каждый последующий пункт конфигурирует получение документов. Каждый пункт получения в DocForDoc конфигурирует как документы, представленные данной конфигурацией DocExt ( целевые документы ), будут получены из какого-либо другого документа ( исходный документ ).

К пункту получения DocForDoc можно обращаться из пунктов конфигураций Doc и DocUpDoc и нескольких публичных функций библиотеки DocExtLib.

### [Формат пункта](#)

Пункт значений по умолчанию конфигурации DocForDoc является первым пунктом из всех. Этот пункт определяет значения по умолчанию для получения целевых документов в каком-либо пункте DocForDoc.

Формат пункта значений по умолчанию:

**APP[ наименование ] VIEW[ наименование ] PICK[ наименование ] FOLDER[ наименование ]**

Описание формата:

**APP[ наименование ]** - наименование приложения ( в проекте ), из которого будет получены целевые документы. Проекты и их приложения определяются конфигурациями типа Project.

**VIEW[ наименование ]** - наименование представления, из которого будут получены целевые документы.

**PICK[ наименование ]** - наименование представления, из которого будут выбираться целевые документы пользователем.

**FOLDER[ наименование ]** - наименование папки ( не представления ), из которого будут выбраны целевые документы пользователем. Папка будет использована вместо представления ( тег PICK ) либо чтобы выбрать документы по более чем одному ключу, либо когда проверяется условие для целевых документов с помощью тегов TRG или BOTH.

Пункты получение документов должны размещаться под пунктом значений по умолчанию ( первым ) в конфигурации DocForDoc. Вводите каждый новый пункт конфигурации на новой строке.

Формат пункта получения документов:

**наименование [ описание ] SRC{ условие } KEY{ формула } TRG{ условие } BOTH{ условие } APP[ наименование ] PRJ[ наименование ] VIEW[ наименование ] PICK[ наименование ] FOLDER[ наименование ] CAT{ формула }**

Описание формата:

**наименование** - уникальное наименование пункта в конфигурации ( без пробелов ).

Рекомендуется использовать короткое наименование и начать его с текста, соответствующего исходным документам, например - наименование их формы. Полное наименование пункта конфигурации, с помощью которого можно обращаться к пункту, будет являться объединением наименований конфигурации и его пункта через символ "4". Таким образом, полное наименование пункта будет содержать информацию как о целевых документах, так и об исходном документе. Например, если конфигурация DocExt называется "Contract" ( целевые документы ) и пункт конфигурации DocForDoc называется "Company" ( исходный документ ), то полное наименование пункта конфигурации будет "Contract4Company". Чтобы обратиться к пункту конфигурации DocForDoc откуда-либо, Вы должны всегда использовать его полное наименование.

**[ описание ]** - текст описания для пункта. Например - [ Договоры для Компании ].

**SRC{ условие }** - определяет когда выполнять получение документов. Условие проверяется для исходного документа.

**KEY{ формула }** - вычисляет ключ, чтобы получить или выбрать целевые документы в представлении. Результатом должен быть текст. Формула вычисляется на исходном документе.

**TRG{ условие }** - определяет когда целевой документ остается в полученном списке документов. Условие проверяется для каждого целевого документа, полученного из представления ( тег VIEW ) по ключу ( тег KEY ).

**BOTH{ условие }** - определяет когда целевой документ остается в полученном списке документов. Условие проверяется для каждого целевого документа, полученного из представления ( тег VIEW ) по ключу ( тег KEY ) и затем проверенного условием ( тег TRG ), и для исходного документа одновременно. Внутри формулы пишите теги SRC[ наименование ] и TRG[ наименование ], чтобы использовать значение поля с указанным наименованием в исходном и целевом документе соответственно. Результатом, полученным тегом SRC или TRG, всегда будет текст, заключенный в кавычки. Если результатом тега будет список, то каждое его значение будет заключено в кавычки с разделением всех значений двоеточием между собой для преобразования этого списка в формат Языка Формул.

**VIEW[ наименование ]** или **VIEW{ формула }** - наименование представления или формула, вычисляющая его. Заменяет наименование, указанное в теге VIEW пункта конфигурации со значениями по умолчанию.

**PICK[ наименование ]** или **PICK{ формула }** - наименование представления или формула, вычисляющая его. Заменяет наименование, указанное в теге PICK пункта конфигурации со значениями по умолчанию.

**FOLDER[ наименование ]** или **FOLDER{ формула }** - наименование папки или формула, вычисляющая его. Заменяет наименование, указанное в теге FOLDER пункта конфигурации со значениями по умолчанию.

**CAT{ формула }** - вычисляет единственную категорию для отображения документов в папке. Тег имеет значение только для выбора пользователем документов в папке, полученной тегом FOLDER, когда результатом тега KEY является список.

**APP[ наименование ]** или **APP{ формула }** - наименование приложения ( в проекте ) или формула, вычисляющая его. Заменяет наименование, указанное в теге APP пункта конфигурации со значениями по умолчанию. Проекты и их приложения определяются конфигурациями типа Project.

**PRJ[ наименование ]** или **PRJ{ формула }** - наименование проекта с приложением, определяемым тегом APP. Используйте этот тег только если проект целевых документов отличается от проекта исходного документа. Проекты и их приложения определяются конфигурациями типа Project.

В тегах VIEW, PICK, FOLDER, APP, PRJ со скобками { } формула вычисляется на исходном документе. Этими тегами, а также тегом KEY, Вы можете вычислять список в качестве результата, но в таком случае размерности списков либо должны быть равными между собой, либо должны равняться одному.

### **2.2.3 [Тип DocExt - Пункты DocUpDoc](#)**

#### **[Описание](#)**

Пункты конфигурации DocUpDoc являются частью конфигурации DocExt и определяют, как изменения в одном документе будут отражены в других. Кроме мгновенного обновления документов, конфигурация позволяет настроить отложенное обновление посредством сервера с целью обойти уровень доступа пользователей, который недостаточен для изменения документов, но это изменение необходимо выполнить.

Пункты конфигурации DocUpDoc должны вноситься на закладке DocUpDoc конфигурации DocExt. Используйте новую строку для каждого пункта.

К пунктам конфигурации Вы можете обращаться каждый раз, когда документ сохраняется при выполнении какого-либо кода.

Исходным документом для какого-либо пункта конфигурации DocUpDoc является документ, который изменяется в данный момент и представлен конфигурацией DocExt с этим пунктом DocUpDoc. Целевыми документами являются документы, в которых отражаются изменения, сделанные в исходном документе.

Обновление документов конфигурацией DocUpDoc вызывает цепную реакцию. Таким образом, если изменения исходного документа отражаются в целевом документе с помощью конфигурации DocUpDoc и если этот целевой документ имеет собственную конфигурацию DocUpDoc, то изменения в нем будут также отражены в его целевых документах и так далее.

#### Формат пункта

Формат пункта конфигурации DocUpDoc:

**D2D[ наименование ] D4D[ наименование ] NEW{ условие } DEL[ подтеги ] DELAY{ условие }**

Описание формата:

**D2D[ наименование ]** - наименование конфигурации DocToDoc. Если произойдет изменение в исходном документе, описанное каким-либо пунктом конфигурации DocToDoc, то это изменение будет отражено в целевых документах;

**D4D[ наименование ]** - полное наименование пункта конфигурации DocForDoc, чтобы получать целевые документы для их обновления. Вы можете написать здесь расширенную форму получения документов: **D4D[ SRC{ условие } наименование TRG{ условие } ]** - подробнее об этом формате читайте в конце темы справки - Конфигурация DocExt\Doc;

**NEW** или **NEW{ условие }** - означает, что если новый исходный документ сохраняется в первый раз, то данные из него будут добавлены в целевые документы. Если тег написан с условием, то оно будет проверяться для каждого целевого документа.

**DEL[ подтеги ]** - часть пункта конфигурации определяет действие с целевыми документами, когда исходный документ удаляется. Этот тег содержит внутри подтеги:

**FAIL** или **FAIL{ условие }** - означает сбой удаления, если результат проверки условия для какого-либо целевого документа любого уровня в цепочке обновлений является @True (1). Напишите тег без условия, чтобы просто наличие целевых документов вызывало сбой удаления.

**STOP** или **STOP{ условие }** - работает как FAIL, но проверяются целевые документа только первого уровня в цепочке обновлений.

**ERASE** или **ERASE{ условие }** - определяет когда значения исходного документа будут очищены в целевых документах.

**ALSO** или **ALSO{ условие }** - определяет когда целевые документы будут также удалены вместе с исходным документом.

Если Вы хотите выполнять какое-либо действие подтега безусловно, то просто напишите этот подтег без скобок { }. Подтеги при удалении исходного документа будут применяться в той же последовательности как они перечислены в данном описании тега DEL до первого результата проверки условия подтега @True (1).

**DELAY** или **DELAY{ условие }** - определяет когда целевые документы будут обновлены посредством сервера. Это может быть полезно, если пользователь, который изменяет исходный документ, не имеет доступа выполнять обновление по пункту конфигурации DocUpDoc в целевых документах. Напишите только тег DELAY без скобок { }, чтобы выполнять отложенное обновление безусловно.

## 2.3 Тип DocToDoc

### Описание

Конфигурация DocToDoc определяет часть данных, которая будут передаваться из исходного документа в целевой. С помощью этой конфигурации Вы можете управлять правилами, когда и какие данные будут переданы между двумя документами. Кроме того, Вы можете добавлять поствычисление для целевого документа, которое будет выполнено сразу после передачи данных. Конфигурация также поддерживает множественную передачу данных multi-to-one, которая означает, что значения исходного документа размещаются в списке значений целевого документа.

Конфигурация DocToDoc состоит из её пунктов. Пишите каждый пункт на новой строке. Пункт конфигурации определяет данные для передачи ( пункт передачи ) или вызывает другую конфигурацию DocToDoc ( ссылочный пункт ). Пункты обоих типов могут быть написаны в конфигурации одновременно.

Вы можете обращаться к конфигурации DocToDoc из пункта конфигурации DocUpDoc и некоторый публичных функций библиотеки DocExtLib.

### Добавление

Чтобы создать конфигурацию DocToDoc выполните:

- Откройте представление **Конфигурации \ DocToDoc** и в его панели действий нажмите **Создать конфигурацию DocToDoc**;
- На открывшейся форме в поле **Наименование** введите наименование конфигурации без пробелов. Это должно быть уникальное наименование среди всех конфигураций DocToDoc. С помощью указанного наименования Вы сможете обращаться к данной конфигурации из программного кода и других конфигураций;
- Справа укажите, включена конфигурация или нет. К конфигурации, не отмеченной как включенная, невозможно будет обратиться вообще;

- В поле **Описание** введите текст описания конфигурации. Например - Из Компании в Договор. Это справочная информация, которая не будет использована где-либо;
- В полях **Версия** укажите номер и дату версии конфигурации. Это справочная информация, которая не будет использована где-либо.

### Формат пункта

Пункт может состоять из префиксной, основной и целевой частей. Но только основная часть всегда требуется.

[ Префиксная часть ] Основная часть [ Целевая часть ]

#### **Префиксная часть**

Префиксная часть определяет, являются ли передаваемые данные значениями по умолчанию, является ли передача multi-to-one или добавляет другую конфигурацию DocToDoc в текущую. Определяется это тегами:

**DFT** – означает, что передаваемые данные являются значениями по умолчанию и не будут обновляться после передачи, если исходный документ будет изменен или удален. Это относится к обновлению документа с помощью пункта конфигурации DocUpDoc, где данная конфигурация DocToDoc используется. В программном коде Вы можете контролировать передачу по данному пункту с помощью входного параметра используемой функции.

**MLT** - означает, что значение исходного документа размещается в списке значений целевого документа ( передача multi-to-one ). Используйте этот тег, если хотите размещать значения из более чем одного документа в одном целевом документе. Вы можете написать этот тег в более чем одном пункте конфигурации.

**KEY** - имеет значение только при использовании в этом же пункте конфигурации тега MLT. Означает, что значение исходного документа является ключевым среди всех значений, передаваемых в список значений целевого документа. Все вместе пункты передачи в конфигурации с тегом KEY должны обеспечить уникальность передаваемых ими значений в целевом документе при добавлении или обновлении. Иначе конфигурация DocToDoc не будет обработана и будет показано окно сообщения об ошибке.

**D2D** - означает, что другая конфигурация DocToDoc будет добавлена в данную конфигурацию. Если Вы также используете в пункте теги DFT, MLT, KEY, то они будут применены ко всем пунктам добавляемой конфигурации DocToDoc. В добавляемой конфигурации Вы также можете использовать тег D2D, чтобы добавить другую конфигурацию и так далее.

#### **Основная часть**

Основная часть пункта конфигурации определяет какие данные исходного документа ( источник для краткости ) и в какое поле целевого документа ( получатель для краткости ) будут переданы. Используйте один из возможных формат, когда пишете основную часть пункта конфигурации DocToDoc:

**наименование** - означает наименование идентичных полей источника и получателя или наименование другой конфигурации DocToDoc, если написан тег D2D. Просто напишите наименование поля или конфигурации.

**SRC{ формула }** - означает вычисление наименования идентичных полей источника и получателя или наименование другой конфигурации DocToDoc, если написан тег D2D. Напишите формулу, которая будет вычислена на источнике, чтобы получить наименования поля или конфигурации.

**TRG{ формула }** - означает вычисление наименования идентичных полей источника и получателя или наименование другой конфигурации DocToDoc, если написан тег D2D. Напишите формулу, которая будет вычислена на получателе, чтобы получить наименования поля или конфигурации.

**BOTH{ формула }** - означает вычисление наименования идентичных полей источника и получателя или наименование другой конфигурации DocToDoc, если написан тег D2D. Напишите формулу, которая будет вычислена одновременно на источнике и на получателе, чтобы получить наименование поля или конфигурации. Внутри формулы вводите теги SRC[ наименование ] и TRG[ наименование ], чтобы использовать значения полей с указанными наименованиями в источнике и в получателе соответственно. Результат, полученный тегом SRC или TRG, будет всегда заключен в кавычки. Если результатом является список, то каждое его значение будет заключено в кавычки с разделением значений списка двоеточием между собой, чтобы представить их в формате текстового списка Языка Формул.

Все теги, описанные ниже, могут быть использованы, только если тег D2D не указан в пункте конфигурации.

**наименование1** или **{ формула } >> наименование2** - означает, что поля источника и получателя не идентичны или вместо наименования поля источника используется формула. Впишите наименование поля или формулу ( в скобках ) источника слева от ">>" и наименование поля получателя справа.

**наименование** или **{ формула } >> SRC{ формула }** - означает, что поле получателя будет вычислено на источнике. Напишите наименование поля или формулу ( в скобках ) источника слева от ">>" и формулу в теге SRC{ } справа, которая будет вычислена на источнике, чтобы получить наименование поля получателя.

**наименование** или **{ формула } >> TRG{ формула }** - означает, что поле получателя будет вычислено на самом получателе. Напишите наименование поля или формулу ( в скобках ) источника слева от ">>" и формулу в теге TRG{ } справа, которая будет вычислена на получателе, чтобы получить наименование его поля.

**наименование** или **{ формула } >> BOTH{ формула }** - означает, что наименование поля получателя будет вычислено одновременно на источнике и на получателе. Напишите наименование поля или формулу ( в скобках ) источника слева от ">>" и формулу в теге BOTH{ } справа, которая будет вычислена одновременно на источнике и на получателе, чтобы получить наименование поля получателя. Как работает тег BOTH описано немного выше.

Во всех случаях, когда используется формула, чтобы вычислить наименование поля получателя или другой конфигурации DocToDoc, пустой текст в качестве результата означает, что данный пункт конфигурации не будет применен.

### **Целевая часть**

После основной часть пункта DocToDoc Вы можете написать целевую часть в следующем формате:

**D4D[ наименование ]** - означает, что вместо измененного значения исходного документа будут использованы список значений из документов, полученных для целевого документа по пункту конфигурации DocForDoc с указанным наименованием. Вы также можете написать тег D4D{ формула }, чтобы вычислять пункт конфигурации DocForDoc на целевом документе. Если исходный документ не является новым ( не сохраненным ), он должен быть среди полученных, иначе это вызовет ошибку. Вы можете оставить полученный список значений как результат или обработать его с помощью тега { формула }, написанного далее. Если тег D2D был написан в префиксной части пункта конфигурации, то тег D4D будет применен ко всем пунктам добавляемой конфигурации DocToDoc.

**{ формула }** - означает, что значением поля получателя будет являться результат вычисления формулы на нем. Но это случится только если значение источника, определенное в данном пункте конфигурации, было изменено. Имеет значение только для пункта конфигурации без тега D2D. Передача значения источника в поле получателя перед вычислением формулы будет выполняться только при наличии наименования этого поля в формуле ( с учетом регистра ).

**FAIL{ условие }** - определяет когда результат передачи вызовет ошибку. Напишите в формуле наименование поля, измененного данным пунктом конфигурации, и другие наименования полей получателя, если это потребуется. Результат формулы @True (1) или любой непустой текст вызовет ошибку с окном сообщения и остановит передачу по конфигурации. Если конфигурация DocToDoc вызывается из пункта DocUpDoc конфигурации DocExt перед сохранением исходного документа, то процесс обновления связанных целевых документов будет остановлен и метод Update ( класс DocExt ) вернет False (0).

## 2.4 Тип Project

### Описание

Конфигурация Project определяет приложение или группу приложений, объединенных по какому-либо смыслу ( например - по году, по наименованию проекта или по тому и другому ), - далее проект-образующие приложения. Кроме этого, конфигурация Project должна содержать ссылки на все приложения, к которым возможно обращение из проект-образующих приложений, - далее проект-связанные приложения.

Первый пункт конфигурации Project определяет значения по умолчанию и каждый последующий либо определяет проект-образующее приложение, либо ссылается на проект-связанное приложение в другой конфигурации Project. Используйте новую строку для каждого пункта при написании конфигурации.

К приложению, описанному пунктом конфигурации Project, можно обратиться из пункта конфигурации DocForDoc и некоторых публичных функций библиотеки DocExtLib.

### Добавление

Чтобы создать конфигурацию Project выполните:

- Откройте представление **Конфигурации \ Project** и нажмите **Создать конфигурацию Project** в его панели действий;

- В открывшейся форме в поле **Наименование** впишите наименование для конфигурации без пробелов. Это должно быть уникальное наименование среди всех конфигураций Project. С помощью указанного наименования Вы сможете обращаться к данной конфигурации из программного кода или других конфигураций;
- Справа укажите, является конфигурация включенной или нет. К конфигурации, не отмеченной как включенная, невозможно будет обратиться вообще;
- В поле **Описание** напишите текст с описанием конфигурации. Например, Договоры 2015. Описание можете показано в окнах сообщения при выполнении программного кода;
- В полях **Версия** укажите номер и дату версии конфигурации. Это справочная информация, которая не будет использована где-либо.

#### Формат пункта

Используйте первый пункт в конфигурации Project чтобы установить значения по умолчанию для проект-образующих приложений в этой конфигурации. Если Вы не напишите пункт значений по умолчанию отдельно, то они будут определяться первый пунктом с приложением в конфигурации.

Формат пункта конфигурации со значениями по умолчанию:

**SRV[ наименование ] FOLDER[ наименование ]**

Описание формата:

**SRV[ наименование ]** - сокращенное наименование сервера по умолчанию;

**FOLDER[ наименование ]** - наименование папки относительно директории данных клиента Notes или сервера Domino.

Для пункта конфигурации, определяющего какое-либо проект-формирующее приложение, используйте формат:

**наименование [ описание ] SRV[ наименование ] FILE[ наименование ] RID[ идентификатор ]**

Описание формата:

**наименование** - уникальное наименование для приложения в конфигурации. С помощью этого наименования Вы сможете обращаться к приложению, описываемому в данном пункте конфигурации, из других конфигураций или программного кода.

**[ описание ]** - описание приложения в проекте. Описание может быть показано в окне сообщения при выполнении кода.

**SRV[ наименование ]** - сокращенное наименование сервера с приложением. Не пишите этот тег, чтобы использовать наименование сервера по умолчанию, указанное в первом пункте конфигурации.

**FILE[ наименование ]** - наименование файла приложения. Напишите полное наименование файла относительно директории данных клиента Notes / сервера Domino или напишите символ "\" в начале, чтобы ввести наименование файла относительно папки по умолчанию, указанной в первом пункте конфигурации.

**RID[ идентификатор ]** - восьмиразрядный идентификатор реплики приложения. Введите этот тег для каждого приложения или, вместо этого, нажмите кнопку "Установить ID реплики" внизу перед сохранением конфигурации, чтобы добавить тег RID с идентификатором реплики для всех приложений в конфигурации, где тег RID не был введен.

### **3 Библиотека DocExtLib**

#### **Описание**

DocExtLib является единственной библиотекой Lotus Script и она должна быть вставлена во все приложения, в которых Вы планируете использовать Конфигурации Doc Ext ( DEC ).

Библиотека работает с документами конфигурации, внесенными в приложение DEC администратором.

Библиотека содержит основной класс DocExt, код его обработки и несколько полезных публичных функций, которые могут быть вызваны из приложений, в которые библиотека DocExtLib была предварительно вставлена.

#### **3.1 Класс DocExt**

##### **Описание**

Класс DocExt является единственным публичным классом библиотеки DocExtLib. Объект класса неразрывно связан с каким-либо документом в приложении. Просто предоставьте этот документ при создании объекта DocExt. Объект класса работает с документом конфигурации типа DocExt, именованным как форма документа, с которым этот объект связан.

##### **Создание объекта**

Вы должны создать объект DocExt, прежде чем документ будет изменен. Таким образом, для UI документа, изменяемого пользователем посредством его формы, создание объекта должно быть написано в событии PostOpen формы. Для back-end документа напишите создание объекта DocExt для него перед всеми изменениями этого документа в программном коде.

Определение объекта:

```
dim doc as NotesDocument
```

```
dim ext as DocExt
```

Простое создание:

```
set ext = new DocExt( doc, "" )
```

Условное создание:

```
if not DEdocLocked( doc ) then
```

```
    set ext = new DocExt( doc, "" )
```

```
end if
```

При условном создании объекта ( *ext* ), документ ( *doc* ) проверяется на его блокировку конфигурацией DocExt\Doc. Если документ не заблокирован, то функция DEdocLocked возвращает результат False (0).

Типичный программный код, который следует писать в событии PostOpen формы:

```
if source.isNewDoc then
    set ext = new DocExt( doc, "" )
elseif source.editMode then
    source.editMode = not DEdocLocked( doc )
    if source.editMode then
        set ext = new DocExt( doc, "" )
    end if
end if
```

### Обработка объекта

Обработка объекта DocExt выполняет все проверки связанного с объектом документа, предусмотренные его конфигурацией DocExt\Doc, и затем, если проверки были успешно пройдены, обновляет все связанные документы с журнализацией изменений.

Чтобы вызвать обработку объекта DocExt в Вашем коде, необходимо вызвать метод объекта под названием *Update*. Метод необходимо вызывать, прежде чем документ будет сохранен. Таким образом, для UI документа, изменяемым пользователем посредством его формы, необходимо вызывать обработку объекта DocExt в событии QuerySave формы. Для back-end документа необходимо вызывать обработку объекта DocExt сразу перед сохранением документа в программном коде.

Метод *Update* возвращает результат: True (-1) or False (0). Результат True означает, что документ успешно прошел все проверки, и связанные документы были обновлены. Проверки определяются пунктами Doc, а обновления определяются пунктами DocUpDoc конфигурации DocExt, связанной с документом по наименованию формы.

Напишите код в событии формы QuerySave, чтобы вызвать обработку объекта DocExt ( *ext* ) при сохранении UI документа:

```
continue = false
continue = ext.update( "" )
```

Напишите код, чтобы вызвать обработку объекта DocExt ( *ext* ) при сохранении back-end документа ( *doc* ):

```
if ext.update( "" ) then doc.save( true, false )
```

## **3.2 Публичные функции**

### Описание

В библиотеке DocExtLib имеется несколько публичных функций, которые могут быть вызваны из Вашего кода для обработки документов в приложениях. Как правило, какая-либо такая функция обращается к одному или более документам конфигурации в приложении DEC, чтобы выполнить действия этой функции. Все публичные функции библиотеки начинаются с двух символов - DE.

### **3.2.1 Функция DEdocDescr**

Получает описание документа, состоящее из его типа и вычисляемого наименования.

#### **Синтаксис**

DEdocDescr( doc, docTypeNum% )

#### **Аргументы**

doc - NotesDocument, входной. Документ, чье описание должно быть получено.

docTypeNum% - integer, входной. Номер падежа получаемого типа. Допустимые номера: 1 - Именительный, 2 - Родительный, 6 - Предложный.

#### **Возвращает**

String.

#### **Использует конфигурации**

DocExt\Doc, названную как форма документа.

### **3.2.2 Функция DEdocID**

Получает идентификатор документа в таком же формате, который используется в качестве тега документа в списке документов, получаемого как результат различных публичных функций библиотеки DocExtLib.

#### **Синтаксис**

DEdocID( doc )

#### **Аргументы**

doc - NotesDocument, входной. Документ, чей идентификатор получается.

#### **Возвращает**

String.

#### **Использует конфигурации**

Нет.

### **3.2.3 Функция DEdocLocked**

Проверяет документ на его блокировку. Блокировка означает, что документ не может быть изменен или удален. Блокировка документа определяется тегами LOCK, написанными в конфигурации DocExt\Doc, названного как форма этого документа. Следует вызывать данную функцию при создании объекта класса DocExt для документа до его изменения в Вашем программном коде. Более детально об этом читайте в теме справки - Создание объекта класса DocExt.

### **Синтаксис**

DEdocLocked( doc )

### **Аргументы**

doc - NotesDocument, входной. Документ, чья блокировка проверяется.

### **Возвращает**

Boolean. Если документ заблокирован, то функция вернет результат True (-1). Иначе - результат False (0).

### **Использует конфигурации**

DocExt\Doc, названную как форма документа.

### **3.2.4 [Функция DEgetApp](#)**

Получает приложение.

### **Синтаксис**

DEgetApp( prjName\$, appName\$, silent, db )

### **Аргументы**

prjName\$ - string, входной. Наименование конфигурации Project, где получаемое приложение описано. Пустая строка означает конфигурацию Project, где описано текущее приложение, из которого вызывается данная функция.

appName\$ - string, входной. Наименование приложения в конфигурации Project.

silent - boolean, входной. Значение True (-1) означает, что если приложение не было получено, то это не вызовет отображение окна сообщение с ошибкой.

db - NotesDatabase, выходной. Приложение, которое должно быть получено.

### **Возвращает**

Boolean. Результат True (-1) означает, что приложение было получено и False (0), что нет.

### **Использует конфигурации**

Project.

### **3.2.5 [Функция DEgetAppName](#)**

Получает наименование приложения из конфигурации Project, в которой оно было описано.

### **Синтаксис**

DEgetAppName( db )

### **Аргументы**

db - NotesDatabase, входной. Приложение, наименование которого должно быть получено. Если будет передан Nothing, то будет получено наименование текущего приложения, из которого вызывается функция.

### **Возвращает**

String. Если конфигурация Project с описанием приложения не была получена, то функция вернет пустую строку.

### **Использует конфигурации**

Project.

### **3.2.6 Функция DEgetC4D**

Получает целевые документы для исходного документа.

### **Синтаксис**

DEgetC4D( D4Dnames\$, trgCond\$, srcDoc, silent, trgDocs )

### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocForDoc, с помощью которого должны быть получены целевые документы. Подробнее об этом наименовании читайте в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать одновременно несколько пунктов конфигурации, перечислите их наименования в строке через пробел.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул для отбора целевых документов, полученных с помощью пункта конфигурации DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены целевые документы.

silent - boolean, входной. Значение True (-1) означает, что, если целевые документы не были получены, то окно сообщение об этом не будет отображено.

trgDocs - NotesDocument List, выходной. Целевые документы, которые должны быть получены. Список может содержать документы из различных приложений.

### **Возвращает**

Boolean. Результат True (-1), если хотя бы один целевой документ был получен, и False (0), если нет.

### **Использует конфигурации**

DocExt\DocForDoc.

### **3.2.7 Функция DEgetD4D**

Получает единственный целевой документ для исходного документа.

### **Синтаксис**

DEgetD4D( D4Dname\$, trgCond\$, srcDoc, silent, trgDoc )

### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocForDoc, с помощью которого должен быть единственный целевой документ. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобразить единственный целевой документ из полученных по пункту конфигурации DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого должен быть получен единственный целевой документ.

silent - boolean, входной. Значение True (-1) означает, что, если не был получен ровно один целевой документ, то окно сообщения об этом не будет показано.

trgDoc - NotesDocument, выходной. Единственный целевой документ, который должен быть получен.

#### **Возвращает**

Boolean. Результат True (-1) означает, что был получен ровно один целевой документ, и False (0), если ни одного или более чем один.

#### **Использует конфигурации**

DocExt\DocForDoc.

### **3.2.8 [Функция DEgetPrjName](#)**

Получает наименование конфигурации Project, в которой описано данное приложение.

#### **Синтаксис**

DEgetPrjName( db )

#### **Аргументы**

db - NotesDatabase, входной. Приложение, для которого должно быть получено наименование конфигурации Project. Если передается Nothing, то будет получено наименование с описанием текущего приложения, из которого вызывается функция.

#### **Возвращает**

String. Если конфигурация Project с описанием приложения не была найдена, то функция вернет пустую строку.

#### **Использует конфигурации**

Project.

### **3.2.9 [Функция DEgetSrcC4D](#)**

Получает документы ( далее источники данных ) для исходного документа и передает данные из них в целевой документ. Объединяет последовательные вызовы двух функций: DEgetC4D и DEprD2D.

#### **Синтаксис**

DEgetSrcC4D( D4Dname\$, D4DtrgCond\$, D4DsrcDoc D4Dsilent, D2Dnames\$, D2Dprocess%, D2DtrgDoc )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocForDoc, с помощью которого должны быть получены источники данных. Читайте подробнее об этом наименовании в теме

справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

D4DtrgCond\$ - string, входной. Дополнительное условие на Языке Формул для отбора документов, полученных пунктом конфигурации DocForDoc.

D4DsrcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены источники данных.

D4Dsilent - boolean, входной. Значение True (-1) означает, что, если не были получены источники данных, то окно сообщения об этом не будет показано.

D2Dname\$ - string, входной. Наименование конфигурации DocToDoc, с помощью которой данные из полученных источников данных должны быть переданы в целевой документ. Чтобы использовать несколько конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входной. Номер процесса передачи: 1 - добавление данных, 3 - очищение данных. Читайте подробнее об этом номере в теме справки - Функция DEprD2D.

D2DtrgDoc - NotesDocument, изменяемый. Целевой документ, в который будет переданы данные из полученных источников данных. Функция может изменить данный документ, но не сохраняет его.

#### **Возвращает**

Boolean. Результат True (1) означает, что источники данных были получены, и какие-либо их данные были переданы в целевой документ, и False (0), если хотя бы одно из этого не произошло.

#### **Использует конфигурации**

DocExt\DocForDoc, DocToDoc.

### **3.2.10 [Функция DEgetSrcD4D](#)**

Получает единственный документ ( далее источник данных ) для исходного документа и передает данные из него в целевой документ. Используйте функцию вместо последовательных вызовов двух других функций - DEgetD4D и DEprD2D, чтобы сократить количество строк программного кода и исключить присутствие источника данных в нем.

#### **Синтаксис**

DEgetSrcD4D( D4Dname\$, D4DtrgCond\$, D4DsrcDoc D4Dsilent, D2Dnames\$, D2Dprocess%, D2DtrgDoc )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должен быть получен источник данных. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

D4DtrgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобразить источник данных из полученных с помощью пункта конфигурации DocExt\DocForDoc.

D4DsrcDoc - NotesDocument, входной. Исходный документ, для которого должен быть получен источник данных.

D4Dsilent - boolean, входной. Значение True (-1) означает, что, если источник данных не был получен, что окно сообщения об этом не будет показано.

D2Dname\$ - string, входной. Наименование конфигурации DocToDoc, с помощью которой данные из полученного источника данных должны быть переданы в целевой документ. Чтобы использовать несколько конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входной. Номер процесса передачи: 1 - Добавление данных, 3 - Очищение данных. Также предусмотрены другие значения. Читайте подробнее об этом номере в теме справки - Функция DEprD2D.

D2DtrgDoc - NotesDocument, изменяемый. Целевой документ, в который должны быть переданы данные из полученного источника данных. Функция может изменить этот документ, но не сохраняет его.

#### **Возвращает**

Boolean. Результат True (1) означает, что источник данных был получен, и какие-либо его данные были переданы в целевой документ, и False (0), что хотя бы одно из этого не произошло.

#### **Использует конфигурации**

DocExt\DocForDoc, DocToDoc.

#### **3.2.11 [Функция DEnotGetC4D](#)**

Проверяет отсутствие целевых документов для исходного документа. Используйте эту функцию вместо функции DEgetC4D, если не хотите иметь список целевых документов в своем программном коде, а только хотите проверить их наличие или отсутствие.

#### **Синтаксис**

DEnotGetC4D( D4Dname\$, trgCond\$, srcDoc, silent )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого будет проверено отсутствие целевых документов. Читайте подробнее об этом наименовании в теме справки Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобрать целевые документы из полученных с помощью пункта конфигурации DocExt\DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого отсутствие целевых документов должно быть проверено.

silent - boolean. Значение True (-1) означает, что если целевые документы были получены, то окно сообщения об этом не будет показано.

#### **Возвращает**

Boolean. Результат True (-1), если целевые документы не были получены, и False (0), если были получены.

#### **Использует конфигурации**

DocExt\DocForDoc.

#### **3.2.12 [Функция DEopenD4D](#)**

Открывает целевой документ, полученный для исходного документа. Если было получено несколько целевых документов, то появится представление для выбора, какие из них должны быть открыты.

#### **Синтаксис**

DEopenD4D( D4Dname\$, trgCond\$, srcDoc, silent )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должен быть получен целевой документ. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобрать целевой документ из полученных с помощью пункта конфигурации DocExt\DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого должен быть получен целевой документ, чтобы затем он был открыт.

silent - boolean, input. Значение True (-1) означает, что, если не были получены документы для открытия, то пустое представление не будет показано.

#### **Возвращает**

Boolean. Результат True (-1) означает, что документы были получены и хотя бы один из них был открыт пользователем.

#### **Использует конфигурации**

DocExt\DocForDoc.

#### **3.2.13 [Функция DEpickC4D](#)**

Получает целевые документы для исходного документа, чтобы затем пользователь выбрал некоторые из них. Отличается от функции DEpickD4D тем, что более одного документа может быть выбрано.

#### **Синтаксис**

DEpickC4D( D4Dname\$, trgCond\$, srcDoc, showEmpty, prompt\$, trgDocs )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должны быть получены целевые документы, чтобы затем пользователь выбрал некоторые из них. Читайте подробнее об этом наименовании в теме справки - Конфигурация

DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобрать целевые документы из полученных с помощью пункта конфигурации DocExt\DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены целевые документы.

showEmpty - boolean, входной. Значение True (-1) означает, что если целевые документы не были получены, то представление без документов для выбора не будет показано.

prompt\$ - string, входной. Текст, который должен быть показан при выборе документов. Например – опишите, для чего этот выбор делается.

trgDocs - NotesDocument List, изменяемый. Если целевые документы были выбраны пользователем, то они будут помещены в этот список. Иначе, этот список останется без изменений.

### **Возвращает**

Boolean. Значение True (-1) означает, что целевые документы были получены и затем пользователь выбрал некоторые из них, и False (0), что хотя бы одно из этого не произошло.

### **Использует конфигурации**

DocExt\DocForDoc.

### **3.2.14 [Функция DEpickD4D](#)**

Получает целевые документы для исходного документа, чтобы затем пользователь выбрал единственный из них. Отличается от функции DEpickC4D тем, что только один документ может быть выбран.

### **Синтаксис**

DEpickD4D( D4Dname\$, trgCond\$, srcDoc, showEmpty, prompt\$, trgDoc )

### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должны быть получены целевые документы, чтобы затем пользователь выбрал единственный из них. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

trgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобрать целевые документы из полученных с помощью пункта конфигурации DocExt\DocForDoc.

srcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены целевые документы.

showEmpty - boolean, входной. Значение True (-1) означает, что если целевые документы не были получены, то представление без документов для выбора не будет показано.

prompt\$ - string, входной. Текст, который должен быть показан при выборе документа. Например – опишите, для чего этот выбор делается.

trgDoc - NotesDocument, изменяемый. Если целевой документ был выбран пользователем, то он будет присвоен переменной. Иначе, эта переменная останется без изменений.

#### **Возвращает**

Boolean. Значение True (-1) означает, что целевые документы были получены и пользователь выбрал единственный из них, и False (0), что хотя бы одно из этого произошло.

#### **Использует конфигурации**

DocExt\DocForDoc.

### **3.2.15 Функция DEpickDocs**

Позволяет пользователю выбрать документы из предоставленных для исходного документа. Отличается от функций DEpickC4D и DEpickD4D тем, что документы для выбора предоставляются в функцию, а не получают с её помощью.

#### **Синтаксис**

DEpickDocs( D4Dname\$, srcDoc, showEmpty, multi, prompt\$, trgDocs )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которой предоставленные документы должны быть выбраны пользователем. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

srcDoc - NotesDocument, входной. Исходный документ для пункта конфигурации DocExt\DocForDoc.

showEmpty - boolean, входной. Значение True (-1) означает, что даже если документы для выбора не были предоставлены, то представление также будет показано.

multi - boolean, входной. Значение True (-1) означает, что более одного документа может быть выбрано пользователем.

prompt\$ - string, входной. Текст, который должен быть показан при выборе. Например – опишите, для чего делается этот выбор.

trgDocs - NotesDocument List, изменяемый. В функцию передается список документов, из которых пользователь должен сделать выбор. Если пользователь выбрал документы из этого списка, то список предоставленных документов будет заменен на список выбранных.

#### **Возвращает**

Boolean. Значение True (-1) означает, что хотя бы один документ был выбран пользователем, и False (0), что нет.

#### **Использует конфигурации**

DocExt\DocForDoc.

### 3.2.16 Функция DEpickSrcC4D

Получает документы для исходного документа, чтобы пользователь выбрал некоторые из них, и затем передает данные из выбранных документов (далее источники данных) в целевой документ. Отличается от функции DEpickSrcD4D тем, что может быть выбран более чем один документ.

#### **Синтаксис**

DEpickSrcC4D( D4Dname\$, D4DtrgCond\$, D4DsrcDoc, showEmpty, prompt\$, D2Dname\$, D2Dprocess%, D2DtrgDoc )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должны быть получены документы для выбора. Затем пользователь выберет некоторые из них в качестве источников данных. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

D4DtrgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобразить документы из полученных с помощью пункта конфигурации DocExt\DocForDoc.

D4DsrcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены выбираемые документы.

showEmpty - boolean, входной. Значение True (-1) означает, что представление будет показано, даже если документы для выбора не были получены.

prompt\$ - string, входной. Текст, который должен быть показан при выборе документов. Например, напишите, для чего делается этот выбор.

D2Dname\$ - string, входной. Наименование конфигурации DocToDoc, с помощью которой данные из выбранных документов должны быть переданы в целевой документ. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входной. Номер процесса передачи: 1 - добавление данных, 3 - очищение данных. Также предусмотрены другие номера. Читайте подробнее об этом номере в теме справки - Функция DEprD2D.

D2DtrgDoc - NotesDocument, изменяемый. Целевой документ, в который должны быть переданы данные из выбранных документов. Функция может изменить документ, но не сохраняет его.

#### **Возвращает**

Boolean. Значение True (-1) означает, что документы были выбраны пользователем, и какие-либо данные из них были переданы в целевой документ, и False (0), что хотя бы одно из этого не произошло.

#### **Использует конфигурации**

DocExt\DocForDoc, DocToDoc.

### 3.2.17 Функция DEpickSrcD4D

Получает документы для исходного документа, чтобы пользователь выбрал единственный из них ( далее источник данных ), и затем передает из выбранного документа в целевой документ. Отличается от функции DEpickSrcC4D тем, что только один документ может быть выбран.

#### **Синтаксис**

DEpickSrcD4D( D4Dname\$, D4DtrgCond\$, D4DsrcDoc, showEmpty, prompt\$, D2Dname\$, D2Dprocess%, D2DtrgDoc )

#### **Аргументы**

D4Dname\$ - string, входной. Наименование пункта конфигурации DocExt\DocForDoc, с помощью которого должны быть получены документы для выбора. Затем пользователь выберет единственный из них в качестве источника данных. Читайте подробнее об этом наименовании в теме справки - Конфигурация DocExt\DocForDoc. Чтобы использовать несколько пунктов конфигураций одновременно, перечислите их наименования в строке через пробел.

D4DtrgCond\$ - string, входной. Дополнительное условие на Языке Формул, чтобы отобразить документы из полученных с помощью пункта конфигурации DocExt\DocForDoc.

D4DsrcDoc - NotesDocument, входной. Исходный документ, для которого должны быть получены выбираемые документы.

showEmpty - boolean, входной. Значение True (-1) означает, что представление будет показано, даже если документы для выбора не были получены.

prompt\$ - string, входной. Текст, который должен быть показан при выборе документа. Например, опишите, для чего этот выбор делается.

D2Dname\$ - string, входной. Наименование конфигурации DocToDoc, с помощью которой данные из выбранного документа должны быть переданы в целевой документ. Чтобы использовать несколько конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входной. Номер процесса передачи: 1 - добавление данных, 3 - очищение данных. Также предусмотрены другие номера. Читайте подробнее об этом номере в теме справки - Функция DEprD2D.

D2DtrgDoc - NotesDocument, изменяемый. Целевой документ, в который должны быть переданы данные из выбранного документа. Функция может изменить документ, но не сохраняет его.

#### **Возвращает**

Boolean. Значение True (-1) означает, что документ был выбран пользователем, и какие-либо данные из него были переданы в целевой документ, и False (0), что хотя бы одно из этого не произошло.

#### **Использует конфигурации**

DocExt\DocForDoc, DocToDoc.

### 3.2.18 [Функция DEprCopyDelay](#)

Выполняет отложенное копирование документа в приложение с помощью сервера. Функция полезна, если пользователь не имеет достаточный уровень доступа к приложению, но документ должен быть добавлен в него. Запрос на копирование документа будет размещен в приложении DEC или другое приложение, настроенное обрабатывать такие запросы на сервере. Читайте подробнее об этом в разделе справки - Отложенное обновление документов.

#### **Синтаксис**

DEprCopyDelay( doc As NotesDocument, dstDb As NotesDatabase, saved As Boolean )

#### **Аргументы**

doc - NotesDocument, входной. Исходный документ, который должен быть скопирован. Это может новый или сохраненный документ.

dstDb - NotesDatabase, входной. Приложение, в которое должен быть скопирован документ.

saved - boolean, входной. Значение True (-1) означает, что копируется сохраненный документ и в результате обработки запроса на копирование будет скопирован сам исходный документ.

Значение False (0) означает, что копируется не сохраненный документ и в этом случае все поля исходного документа будут помещены в запрос на копирование.

#### **Возвращает**

Не возвращает значение.

#### **Использует конфигурации**

Нет.

### 3.2.19 [Функция DEprD2D](#)

Передает данные из исходного документа в целевой документ.

#### **Синтаксис**

DEprD2D( D2Dname\$, D2Dprocess%, srcDoc, trgDoc )

#### **Аргументы**

D2Dname\$ - string, входной. Наименование конфигурации DocToDoc, с помощью которой должны быть переданы данные. Чтобы использовать несколько конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входной. Номер процесса передачи: 1 - добавление данных, 3 - удаление данных.

Также Вы можете использовать расширенные номера вместо 1 и 3 для специальной передачи данных:

11 - добавление данных без значений по умолчанию в конфигурации DocToDoc ( пункты с тегом DFT ). Просто номер 1 добавляет данные со значениями по умолчанию.

12 - добавление данных даже если значения исходного документа ( пункты с тегом MLT ) не могут быть добавлены, так как они уже присутствуют в списке значений целевого документа. Просто номер 1 вызовет ошибку в таком случае.

13 - работает как и номер 12, только если значения исходного документа ( пункты с тегом MLT ) были найдены в списке значений целевого документа, то все значения по умолчанию ( пункты с обоими тегами MLT и DFT ) будут обновлены.

31 - очищение данных со значениями по умолчанию ( пункты с тегом DFT ). Просто номер 3 очищает данных без значений по умолчанию.

32 - очищение данных даже если значения исходного документа ( пункты с тегом MLT ) не были найдены в списке значений целевого документа, то список всех значений целевого документа будет показан, чтобы пользователь выбрал какую позицию в этом списке необходимо очистить.

srcDoc - NotesDocument, входной. Исходный документ, из которого должны быть переданы данные. Это может быть Nothing только для очищения данных, если конфигурации DocToDoc не имеет пунктов, вычисляемых на исходном документе ( тег SRC ).

trgDoc - NotesDocument, изменяемый. Целевой документ, в который должны быть переданы данные. Функция может изменить документ, но не сохраняет его.

#### **Возвращает**

Boolean. Значение True (-1) означает, что какие-либо данные из исходного документа были переданы в целевой.

#### **Использует конфигурации**

DocToDoc.

#### **3.2.20 [Функция DEprD2Ddelay](#)**

Выполняет отложенную передачу данные из исходного документа в целевой документ с помощью сервера. Функция может быть полезной, если у пользователя нет достаточного уровня доступа для изменения документа, но некоторые данные должны быть перенесены в него. Запрос на обновление документа будет помещен в приложение DEC или другое приложение, настроенное обрабатывать такие запросы на сервере. Читайте подробнее об этом в разделе справки - Отложенное обновление документов.

#### **Синтаксис**

DEprD2Ddelay( D2Dname\$, D2Dprocess%, srcDoc, trgDoc )

#### **Аргументы**

D2Dname\$ - string, входящий. Наименование конфигурации DocToDoc, с помощью которой должны быть переданы данные. Чтобы использовать несколько конфигураций одновременно, перечислите их наименования в строке через пробел.

D2Dprocess% - integer, входящий. Номер процесса передачи: 1 - добавление данных, 3 - очищение данных. Читайте подробнее об этом номере в разделе справки - Функция DEprD2D.

srcDoc - NotesDocument, входящий. Исходный документ, из которого должны быть переданы данные. Это может быть Nothing только для очищения данных, если конфигурации DocToDoc не имеет пунктов, вычисляемых на исходном документе ( тег SRC ).

trgDoc - NotesDocument, изменяемый. Целевой документ, в который должны быть переданы данные. Документ может быть изменен с помощью запроса, который будет отправлен на сервер для его обработки. Вы не должны затем сохранять документ в программном коде, чтобы сохранить изменения в нем, так как это будет сделано при обработке запроса на сервере специальным агентом.

#### **Возвращает**

Boolean. Значение True (-1) означает, что некоторые данные из исходного документа будут переданы в целевой с помощью запроса на отложенное обновление документа.

#### **Использует конфигурации**

DocToDoc.

### **3.2.21 [Функция DEprDelay](#)**

Обрабатывает запрос на отложенное обновление документа с помощью сервера. Функция используется в специальном агенте и в действии представления. Агент должен запускаться на сервере в приложении DEC или другом приложении, используемом вместо него. Действие необходимо использовать для понимания какой-либо проблемы при отладке программного кода.

#### **Синтаксис**

DEprDelay( delayDoc )

#### **Аргументы**

delayDoc - NotesDocument, изменяемый. Документ, представляющий запрос на отложенное обновление документа в каком-либо приложении. Читайте подробнее об этом документе в разделе справки - Отложенное обновление документов.

#### **Возвращает**

Не возвращает значение.

#### **Использует конфигурации**

DocExt\DocUpDoc, названную как форма измененного документа.

### **3.2.22 [Функция DEprDelete](#)**

Удаляет документ, если он не заблокирован конфигурацией DocExt\Doc, с удалением или обновлением всех связанных документов и с журнализацией этих действий. Связанные документы определяются конфигурацией DocExt\DocUpDoc ( пункты с тегом DEL ), названной как форма удаляемого документа. Используйте эту функцию с событием QueryDocumentDelete ( Код базы данных ) в Ваших приложениях.

#### **Синтаксис**

DEprDelete( doc )

## Аргументы

doc - NotesDocument, изменяемый. Документ, который удаляется.

## Возвращает

Boolean. Значение True (-1) означает, что документ не заблокирован и был успешно удален, а также были успешно удалены или обновлены все связанные с ним документы.

## Использует конфигурации

DocExt\Doc, DocExt\DocUpDoc.

### 3.2.23 [Функция DEprMove](#)

Перемещает документ и все связанные с ним документы в проект с журнализацией. Документы перемещаются в приложения какого-либо другого проекта. Каждый перемещаемый документ переносится в приложение целевого проекта, которое названо точно также как и текущее приложение документа названо в исходном проекте. Связанные документы определяются конфигурацией DocExt\DocUpDoc ( пункты с тегом DEL[ ALSO ] ), названной как форма перемещаемого документа.

## Синтаксис

DEprMove( doc, trgPrjName\$ )

## Аргументы

doc - NotesDocument, изменяемый. Документ, который должен быть перемещен.

trgPrjName\$ - string, входной. Наименование конфигурации Project с приложением, в которое должен быть перемещен документ.

## Возвращает

Boolean. Значение True (-1) означает, что сам документ и все связанные с ним документы были успешно перенесены в приложения целевого проекта.

## Использует конфигурации

Project, DocExt\DocUpDoc.

### 3.2.24 [Функция DEpromptApp](#)

Позволяет пользователю выбрать приложение из конфигурации Project.

## Синтаксис

DEpromptApp( prjName\$, appName\$, appDescr\$ )

## Аргументы

prjName\$ - string, входной. Наименование конфигурации Project, из которой должно быть выбрано приложение.

appName\$ - string, выходной. Наименование выбранного приложения в конфигурации Project.

appDescr\$ - string, выходной. Описание выбранного приложения. Это текст в скобках [ ], следующий за наименованием приложения в пункте конфигурации Project.

## **Возвращает**

Boolean. Значение True (-1) означает, что приложение было выбрано пользователем.

## **Использует конфигурации**

Project.

## **4 Настройки**

### **4.1 Начало использования приложения**

#### **Размещение приложения**

Для предприятия достаточно использовать единственное приложение DEC, в котором будут размещены конфигурации для всех видов документов из всех приложениях предприятия.

Для определения местоположения приложения DEC используются переменные NOTES.INI: DE\_CONFIG\_SERVER - сервер, где размещено приложение, DE\_CONFIG\_FILE - наименование файла относительно папки данных Notes/Domino или ID реплики приложения.

Изначально предполагается, что приложение будет размещено в папке данных сервера приложений с наименованием файла DEconfig.nsf. Если так и было сделано, то каких-либо других действий не потребуется. Переменные NOTES.INI на компьютерах пользователей будут установлены для использования этого приложения при первом обращении к нему.

Если Вы хотите поместить приложение в собственную папку, то необходимо установить переменные NOTES.INI. Вы можете сделать это с помощью пункта меню Notes: Действия \ Конфигурации Doc Ext \ Установить это приложение для этого компьютера.

#### **Уровень доступа к приложению**

Достаточно установить всем пользователям уровень доступа к приложению "Читатель", чтобы они могли обращаться к конфигурациям при работе с приложениями. Вы можете ограничить доступ к определенной конфигурации с помощью полей Читатели и Авторы в документе конфигурации.

В приложение предусмотрены две роли: [ReadAll] - всегда помещается в поле Читатели и позволяет просмотреть любую конфигурацию, и [EditAll] - всегда помещается в поле Авторы и позволяет редактировать любую конфигурацию.

### **4.2 Журнал обновления документов**

Приложение DEC позволяет регистрировать изменения документов, а также их создания и удаления. Чтобы регистрировать создание и обновление документа Вам необходимо использовать для него объект класса DocExt ( библиотека кода DocExtLib ) в программном коде. Чтобы регистрировать удаление документа, Вам необходимо вставить специальный программный код в событие QueryDocumentDelete раздела приложения "Код базы данных". Поля документа, чьи изменения должны регистрироваться, определяются конфигурацией DocExt\Doc, названной как форма изменяемого документа.

#### **Используемое приложение**

Приложение DEC содержит элементы дизайна для развертывания журнала обновления документов. Вы можете использовать само приложение DEC в качестве журнала, но удобнее использовать вместо него отдельное приложение с некоторыми его элементами дизайна. Вы также сможете использовать это же приложение для обработки отложенного обновления документов.

Вам необходимо описать приложение, выбранное в качестве журнала обновления документов, в конфигурации Project. Это может быть конфигурация, в которой также описаны приложения с документами, чьи обновления должны регистрироваться, или может быть отдельная конфигурация с журналом, на который существуют ссылки из других конфигураций Project. Это определяется тем, сколько журналов Вы хотите использовать. По умолчанию наименование приложения в конфигурации Project, по которому Вы будете обращаться к журналу, определяется переменной DE\_LOG\_APP библиотеки кода DocExtLib с исходным значением "Log". Вы можете заменить это наименование приложения по умолчанию с помощью тега LOG\_APP в конфигурации DocExt\Doc.

### **Элементы дизайна**

В списке перечислены элементы дизайна в приложение DEC, которые требуются для развертывания журнала обновления документов. Если Вы решили использовать отдельное приложение в качестве журнала, то необходимо создать новое приложение и вставить эти элементы дизайна в него.

- Представление "Журнал обновления документов" - для просмотра обновлений документов.
- Форма "Журнал обновления документов - Запись" - отображает детали обновления документа.

### **Запись журнала**

Записью журнала является документ в журнале, который может быть показан с помощью формы "Журнал обновления документа". Такой документ содержит поля об изменении самого документа, а также обо всех обновлениях в других документах, связанных в этом изменении.

- DocFile - наименование файла приложения с измененным документом;
- DocServer - наименование сервера приложения с измененным документом;
- DocUNID - UNID измененного документа;
- Message - текстовый список об изменении документа. Первый элемент в списке содержит информацию о пользователе, который сделал изменения, и о измененном документе. И каждый следующий элемент описывает изменение поля документа;
- Targets - Rich Text, содержит информацию об обновлениях во всех связанных документах, вызванных данным изменением. Кроме текста, поле содержит ссылки на обновленные связанные документов;
- Time - время, когда были сделаны изменения;
- User - пользователь, который сделал изменения.

### **Журнализация в поле документа**

Кроме регистрации обновления документов в приложение, Вы также можете регистрировать их в специальное поле измененного документа. По умолчанию наименование поля определяется переменной DE\_LOG\_FIELD библиотеки кода DocExtLib с начальным значением "Log". Вы можете либо поменять значение этой переменной, либо установить значение, отличное от значения по умолчанию, с помощью тега LOG\_FIELD в конфигурации DocExt\Doc. Это поле должно быть помещено в форму документа со свойством "Разрешить список значений" и разделителем значений "Новая строка", чтобы корректно отображать журнал изменений документа.

### **4.3 Отложенное обновление документов**

Приложение DEC позволяет настроить отложенное обновление документов с помощью сервера. Эта отсрочка может определяться конфигурацией Doc\DocUpDoc и некоторыми специальными функциями библиотеки кода DocExtLib. При сохранении документа и запуске обновления всех связанных с ним документов, если пользователь не имеет доступ на редактирование некоторых документов, но при этом включено отложенное обновление для них, то такие документы будут обновлены с помощью отправки запросов на сервер. По умолчанию эти запросы на обновление документов отправляются в специальную базу данных ( mail-in ) на сервере, где они будут мгновенно обработаны специальным агентом.

#### **Используемое приложение**

Приложение DEC содержит элементы дизайна для развертывания отложенного обновления документов. Для развертывания Вы можете использовать само приложение DEC, но удобнее вместо него использовать отдельное приложение с некоторыми его элементами дизайна. Вы также сможете использовать это же приложение для развертывания журнала обновления документов.

#### **Элементы дизайна**

В списке перечислены элементы дизайна приложения DEC, необходимые для развертывания отложенного обновления документов. Их необходимо скопировать в новое отдельное приложение, если Вы решили использовать его для этой цели вместо DEC.

- Библиотека кода DocExtLib. Будет обрабатывать все запросы на обновление документов.
- Агент "Отложенное обновление документов - Обработка входящего запроса", триггер - "До получения новой почты". Будет мгновенно обрабатывать запрос на обновление документа, полученный как сообщение электронной почты.
- Агент "(Отложенное обновление документов - Обработать запрос)". Выполняется на выбранных в представлении документах. Агент необходимо использовать для ручной обработки запроса на обновление документа, чтобы получить информацию об ошибке, из-за которой он не был обработан автоматически.
- Агент "(Отложенное обновление документов - Установить запрос как необработанный)". Выполняется на выбранных в представлении документах. Агент устанавливает состояние Новый в запросе на обновление документа, чтобы он мог быть обработан повторно.
- Представление "Отложенное обновление документов". Для просмотра запросов на обновление документов и устранения проблем с ними.

- Форма "Отложенное обновление документов - Запрос". Форма отображает запрос на обновление документа.

### **База данных Mail-in**

Чтобы использовать отложенное обновление документов необходимо настроить базу данных mail-in на сервере для получения запросов на обновление документов. По умолчанию используется адрес DEconfig. Чтобы использовать другой адрес, напишите его как значение переменной DE\_DELAY\_MAIL в приложении DEC. Определите, какое приложение Вы будете использовать для развертывания отложенного обновления документов и установите его в качестве базы данных mail-in на сервере. Затем убедитесь, что агент "Отложенное обновление документов - Обработка входящего запроса" включен в этом приложении.

### **Запрос на обновление документа**

Запросом на обновление документа является документ, отправленный в специальное приложение, выбранное обрабатывать эти запросы ( база данных mail-in ). Запрос содержит полную информацию об обновлении документа, включая служебные поля и все поля, которые должны быть обновлены в целевом документе. Запрос может быть просмотрен с помощью формы "Отложенное обновление документа - Запрос" ( DocUpDelay ) в представлении "Отложенное обновление документов" приложения, обрабатывающего запросы.

Служебные поля в запросе на обновление документа начинаются с DEdelay:

- DEdelay - признак того, что документ является запросом;
- DEdelayError - наличие этого поля означает, что запрос был обработан. Любой текст в поле означает, что запрос был обработан с ошибкой и описывает её;
- DEdelayTime – время, когда был создан запрос;
- DEdelayTrgDoc - UNID целевого документа, который должен быть обновлен;
- DEdelayTrgFields - наименования полей в целевом документе. Все перечисленные в списке поля присутствуют в запросе со значениями, которые должны быть установлены в целевом документе;
- DEdelayTrgFile - наименование файла приложения с целевым документом;
- DEdelayTrgServer - наименование сервера приложения с целевым документом.

### **Переменные NOTES.INI на сервере**

Убедитесь, что переменные DE\_CONFIG\_SERVER, DE\_CONFIG\_FILE, DE\_CONFIG\_LANGUAGE установлены в файле NOTES.INI на сервере, на котором размещено приложение для обработки запросов на обновление. Читайте подробнее об этих переменных в разделах справки: Начало использования приложения и Переменные библиотеки кода.

## **4.4 Переменные библиотеки кода**

В приложении DEC используются переменные библиотеки кода DocExtLib, чтобы установить некоторые параметры работы приложения, а также переводить на требуемый язык все сообщения приложения, показываемые при выполнении кода. Значения этих переменных могут

быть переопределены с помощью специальных профилей в приложении. Установка значения для переменной должна удовлетворять синтаксису Lotus Script, иначе это вызовет ошибку, и профиль не будет сохранен. В приложении предусмотрено два типа профилей: Общий и Языковой.

### **Общий профиль**

Профиль содержит переменные, которые влияют на различные аспекты работы приложения DEC. Единственный общий профиль в приложении устанавливает значения для следующих переменных:

- DE\_DEBUG\_D2D - boolean, изначально False. Устанавливает вывод на экран всех изменений полей, вызванных какой-либо конфигурацией DocToDoc. Значение True включает такой вывод на экран, иначе, если значение False, отображаются только изменения полей, перечисленных в конфигурации DocExt\Doc, соответствующей изменяемому документу.
- DE\_DELAY\_MAIL - string, изначально "DEconfig". Адрес электронной почты базы данных mail-in, которая настроена получать и обрабатывать запросы на отложенное обновление документов. Если Вы укажете пустую строку в качестве значения переменной, но в конфигурации DocExt\DocUpDoc установлено отложенное обновление документов, то созданные запросы на обновление документов будут только сохраняться в используемом приложении DEC, без их последующей обработки.
- DE\_LOG\_APP - string, изначально "Log". Наименование приложения ( в конфигурации Project ) по умолчанию, которое используется в качестве журнала обновления документов.
- DE\_LOG\_FIELD - string, изначально "Log". Наименование поля по умолчанию, которое используется для регистрации изменений документа в самом документе.
- DE\_PRINT - boolean, изначально True. Устанавливает вывод на экран любых сообщений при работе приложения. Значение False отключает этот вывод на экран.
- DE\_SHOW\_PROGRESS - long, изначально 10. Минимальное количество связанных документов, обновляемых при изменении исходного документа, при котором должна быть показана полоса прогресса обновления. Установите значение 0, чтобы не показывать полосу прогресса вообще.

Вы можете изменять значения переменных с помощью меню Notes: Действия \ Переменные библиотеки кода \ Общие - Редактировать.

### **Языковой профиль**

Языковой профиль определяет язык сообщений, показываемых при выполнении кода библиотеки DocExtLib. Один языковой профиль содержит переменные, чтобы перевести все сообщения на какой-либо язык. Изначально приложение DEC всегда содержит английский языковой профиль - English. Чтобы добавить новый языковой профиль или отредактировать переменные существующего профиля, выберите в меню Notes: Действия \ Переменные библиотеки кода \ Язык - Редактировать.

### **Установка языка**

Если Вы добавили собственный языковой профиль, то Вы можете установить его по умолчанию для приложения DEC с помощью пункта меню Notes: Действия \ Переменные библиотеки кода \

Язык - Установить по умолчанию для приложения. Этим Вы установите наименование выбранного языка в качестве значения поля Language в общем профиле, и затем это наименование будет распространяться как значение переменной DE\_CONFIG\_LANGUAGE в файле NOTES.INI при первом обращении к приложению DEC на каком-либо компьютере. Чтобы установить альтернативный язык для компьютера, необходимо изменить значение переменной NOTES.INI на наименование языка, для которого создан профиль. Один из способов сделать - выбрать в меню Notes: Действия \ Переменные библиотеки кода \ Язык - Установить для этого компьютера.