Doc Ext Configurations

Application for IBM Lotus Notes and Domino®.

Version 2.0.2

Developer and Administrator Guide

Table of contents

1 Application description

The app Doc Ext Configs (DEC for short) allows to manage documents data in your IBM Lotus Notes® applications centrally with minimum coding in them. Describe in the configs of the app DEC the documents of your apps as well as how a one document is getting others and how is updating them. Moreover, the app grants extended opportunities to log all documents updates in your apps.

The app DEC contains the alone Lotus Script Library for pasting into your apps and admin-defined config documents. Additionally the app includes few design elements to deploy documents' update log and delayed documents update using a server.

Compatibility

The version 2.0.2 of the application was deeply tested and is watched its correct work in IBM Lotus Notes® ver. 6.x.x - 9.x.x with the operating systems MS Windows® ver. XP, Vista, 7, 8, 8.1, 10 in any combinations.

Developer

The author and the proprietor of the app is Vadim Matvienko. For any questions about working the app use the e-mail address support@apps4notes.org. The site of the application: http://www.apps4notes.org.

License

This program is free software and is licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0.txt. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

What's new in version 2.0.2 ?

- Syntax validation of expressions in Formula Language inside configurations has been extended;
- The public function DEopenD4D began to return a result (True or False) as well as can be called in silent mode;
- The document changes now continue to be logged inside it even if log field size exceeds 32KB;
- The global variable DE_PRINT has been added to disable displaying of any messages at application working;
- Description of arguments has been added in all functions of the script library DocExtLib;
- Few little bugs have been fixed.

2 Config Documents

2.1 Description

The config documents store all configuration data about the documents in your apps. There are three types of config documents: DocExt (includes Doc, DocForDoc, DocUpDoc items), DocToDoc and Project.

A config consists from its items. Type each item on new line. An item in depend of config type consists either of name and the next tags or of the tags only. A tag defines either the name of something or action with something. A tag can be typed just in itself or with the brackets [ ], { }, < >. Within the brackets [ ] need type a text to define the tag's value. Within the brackets { } need type a condition or an expression in Formula Language to compute the tag's value or its applying. Type a tag with the brackets < > with an other item's name of the config inside to use the value of the same tag from the specified item. What brackets can be used with a tag see in the tag's description of the help.

If brackets of a tag also is used in value of the tag, you must use the full format of the tag typing with adding the tag name after the brackets: TAG{ ... SUBTAG{ ... } ... }TAG.

To check the syntax of all expressions in Formula Language which have been typed in all tags with the brackets { } of all config items you should press "Check All Syntax" at the top of the config document.

To disable a config's item you should type the REM keyword at the start of the item.

2.2 DocExt type

Description

DocExt is the main type of config documents in the DEC. A DocExt described an one type (form) documents in an app. It consists of three parts: Doc, DocForDoc and DocUpDoc. Each part can be enabled / disabled separately.

Adding

To create the config DocExt perform:

- Open the view **Configs \ DocExt** and in the view action bar press **Create DocExt Config**;
- On the opened form in the field **Name** type the config name without spaces. It can be two names, first is short for your convenience, and second (in new line) is the name matched to the form of documents, described in the config. For example: con and Contract. Both names must be unique among all DocExt configs. You can omit short name but the form name is always required;
- At the right define what parts of the config are enabled: **Doc**, **DocForDoc**, **DocUpDoc**;
- In the field **Description** type extended text about the documents with the form. For example: Company's Contracts. It can be shown for a user when a code running.
- In the fields **Version** set the number and date of the config version. It's help info that will not be used anywhere.

2.2.1 DocExt type - Doc items

Description

The Doc config items are part of the config DocExt and describe the documents' type (form) in itself. For a whole document the config items define its checking for uniqueness, the allowing to be changed or deleted (locking), the logging of changes etc. For a document's field the config item checks it for the filling and valid value.

The Doc config items you should type on the tab Doc of a config DocExt. Use new line for each item. The first items configure a document as a whole and each next item configures a document's field.

To the items Doc you can address each time before the saving a document.

Item's format

Use first items (lines) of a config Doc to configure a document as a whole.

The format of an item configuring a document:

TYPE[name] NAME{ formula } LOG_FIELD[name] LOG_APP[name] LOG_MAIL[subtags] LOCK{ condition } LOCK_D4D[name] UNIQ_D4D[name]

The format description:

TYPE[name] - defines the type name of a document that will be displayed before the document name in a message at a code running. If you omit the tag the document's form name will be used;

NAME{ formula } - computes the name of a document for a message at a code running. If you omit the tag the document's UNID name will be used;

LOG_FIELD[name] - defines the field what the document's updating log will be stored to. If you omit the tag the default field name will be used (is set by the const DE_LOG_FIELD in the Library). If you don't want to log changes to a field use the empty in the tag;

LOG_APP[name] - defines the app (in a project) what a document's update log will be stored to. The projects and their apps are defined in the Project type configs. If you omit the tag the default app name will be used (is set by the const DE_LOG_APP in the Library). If you don't want to log changes to an app use the empty in the tag;

LOG_MAIL[subtags] - defines the notice sending about a document changing. The tag contains the subtags inside:

LOG{ condition } - defines when the notice will be sent. The condition is checked on the log document (but not on changed document) with the single field Message with the content of changes;

DOC{ condition } - defines when the notice will be sent. The condition is checked on the changed document;

TO[address] - defines a notice recipient. You can type TO{ formula } to compute (on the changed document) an one or more e-mail addresses;

SUBJ[text] - defines the message subject. It will be appended to the changed document's name computed by the tag NAME{ formula }. Type the tag SUBJ{ formula } to compute the subject on the changed document.

LOCK{ condition } - defines when a document is locked that forbids its editing and deletion. The condition result @True (1) or not empty text for a document locks it. The text will be added to the message box. You also can use the tag in a field's config item;

LOCK_D4D[name] - works like LOCK{ condition } but a document is locked when the documents by the config item DocForDoc are got. You also can use the tag in a field's config item;

UNIQ_D4D[name] - defines that a document is not unique if the documents by the config item DocForDoc are got. It's recommended use the tag in a field's config item.

The fields' config items should place under the document's (first) config items.

The format of a field's config item:

name [description] LOCK{ condition } LOCK_D4D[name] FILL FILL{ condition } FAIL{ condition } FAIL_D4D[name] VLD{ condition } VLD_D4D[name] UNIQ_D4D[name] UNIQ UNIQ{ condition }

The format description:

name - the name of a field which should be configured;

[description] - the description of the field. It will be used for a code running message and updates logging;

FILL - defines that the field must be filled;

FILL{ condition } - defines when the field must be filled;

FAIL{ condition } - defines when the field value is not valid;

FAIL_D4D[name] - means a value of the field is not valid if the documents by the config item DocForDoc are got;

VLD{ condition } - defines when the field value is valid;

VLD_D4D[name] - means a value of the field is valid only if the documents by the config item DocForDoc are got;

LOCK{ condition } - defines when a document is locked that is can't be edited and deleted;

LOCK_D4D[name] - works like LOCK{ condition } but in the case if the documents by the config item DocForDoc are got;

UNIQ_D4D[name] - means a document is not unique if the documents by the config item DocForDoc are got. Use the tag for the first field taking part in the tag KEY of the config item DocForDoc;

UNIQ - type this tag if the field takes part in all configs items DocForDoc specified in a tag UNIQ_D4D of other config items above;

UNIQ{ condition } - works like UNIQ but only if the result of the condition is @True (1).

For all formulas and conditions in the config tags are expected the results: @True (1), @False (0) or not empty text. If is got an expected result the message box with description and value of the field will be displayed. if the result is not empty text then it will be added to the message box.

The tags LOCK_D4D, FAIL_D4D, VLD_D4D, UNIQ_D4D can be used more than one time in a config item. Moreover, for the tags you can use the extended form of calling a config item DocForDoc: ...[SRC{ condition } name TRG{ condition }] - where condition in the tag SRC will be applied to the source document (which is being checked) and only if the result of the condition will be @True (1) the getting of documents by DocForDoc config item with the specified name will be done. Type a condition in the tag TRG to check it on the got documents - only ones with check's result @True (1) will remain as the result.

2.2.2 DocExt type - DocForDoc items

Description

The DocForDoc config items are part of a config DocExt and describe how one document gets others. The config items grants many capabilities for the easy getting and picking documents from views in your apps.

The DocForDoc config items you should type on tab DocForDoc of a config DocExt. Use new line for each item. The first item configures defaults and each next item configures a documents' getting. Each getting's item in DocForDoc configures how the documents represented by the config DocExt (target documents) will got from an other document (source document).

To a getting item DocForDoc can address from Doc and DocUpDoc config items and few public functions of the library DocExtLib.

Item's format

The defaults item of the config DocForDoc is the first item from all. The item defines defaults for the getting of target documents in a DocForDoc item.

The format of the defaults item:

APP[name] VIEW[name] PICK[name] FOLDER[name]

The format description:

APP[name] - the name of an app (in a project) which the target documents will be got from. The projects and their apps are defined in the configs of the type Project.

VIEW[name] - the name of a view which the target documents will be got from.

PICK[name] - the name of a view which the target documents will be picked from by an user.

FOLDER[name] - the name of a folder (not view) which the target documents will be picked from by an user. The folder will be used instead of the view (tag PICK) either to pick documents by more than one key or when a condition is checked on target documents using tags TRG or BOTH.

The items of documents' getting should be placed under defaults item (the first) in a config DocForDoc. Type each new config item on new line.

The format of an item of documents' getting:

name [description] SRC{ condition } KEY{ formula } TRG{ condition } BOTH{ condition } APP[name] PRJ[name] VIEW[name] PICK[name] FOLDER[name] CAT{ formula }

The format description:

name - unique name of the item in the config (without spaces). It's recommended to use a short name and begin it from text corresponding to the source documents, for example - their form's name. The full name of a config item through which you can address to the item will be the combine of names of the config and of its item through the letter "4". Thus the full name of an item will contain info about both the target documents and a source document. For example, if the config DocExt is named "Contract" (target documents) and an item DocForDoc is named "Company" (source document) the full name of the config item will "Contract4Company". To address to a config item DocForDoc from somewhere, you must always use its full name.

[description] - a text of description for the item. For example - [Contracts for the Company].

SRC{ condition } - defines when to perform the documents' getting. The condition is checked on a source document.

KEY{ formula } - computes the key to get or pick the target documents in the view. The result must be a text. The formula is computed on a source document.

TRG{ condition } - defines when a target document remains in the got ones list. The condition is checked on each target document that was got from view (tag VIEW) by key (tag KEY).

BOTH{ condition } - defines when a target document remains in got ones list. The condition is checked on each target document that has been got from view (tag VIEW) by key (tag KEY) and next checked by condition (tag TRG) and on source document at the same time. Inside the formula type the tags SRC[name] and TRG[name] to use value of the field with specified name in source and target document accordingly. A result got by the tag SRC or TRG will always be a text enclosed in quotes. If a tag result will be a list each its element will be enclosed in quotes with separating the all elements by colon between themselves to transform the list in the format of Formula Language.

VIEW[name] or **VIEW{ formula }** - the name of a view or formula to compute it. Overwrites the name specified in the tag VIEW of the config's item with defaults.

PICK[name] or **PICK{ formula }** - the name of a view or formula to compute it. Overwrites the name specified in the tag PICK of the config's item with defaults.

FOLDER[name] or **FOLDER{ formula }** - the name of a folder or formula to compute it. Overwrites the name specified in the tag FOLDER of the config's item with defaults.

CAT{ formula } - computes single category to display documents in the folder. The tag matters only for the documents picking by a user in the folder got by the tag FOLDER when the result of the tag KEY is list.

APP[name] or **APP{ formula }** - the name of an app, or a formula to compute it. Overwrites the name specified in the tag APP of the config's item with defaults. The projects and their apps are defined in the configs of the type Project.

PRJ[name] or **PRJ{ formula }** - the name of an project with the app defined by the tag APP, or formula to compute it. Use the tag only if the target documents' project differs from source document's project. The projects and their apps are defined in the configs of the type Project.

In the tags VIEW, PICK, FOLDER, APP, PRJ with the brackets { } a formula will be computed on a source document. By all the tags as well as the tag KEY you can compute a list as result but in this case the bounds of the lists either must be equal between themselves or must be equal to one.

2.2.3 DocExt type - DocUpDoc items

Description

The DocUpDoc config items are part of config DocExt and define how changes in one document will be reflected in others. Besides the instant documents update the config allows to set the delayed update through server in order to bypass the users' access level which is insufficient for change the documents but the change need to be performed.

The DocUpDoc config items should type on the tab DocUpDoc of a config DocExt. Use new line for each item.

To the config's items you can address each time when a document is being saved at running of a code.

The source document for a DocUpDoc config item is a document which is now being changed and represented by the DocExt config with the DocUpDoc item. The targets documents are documents which the changes made in the source document are reflected in.

The documents updating by a DocUpDoc config calls the chain reaction. Thus if changes of the source document are reflected in a target document using the DocUpDoc config and if the target document has own DocUpDoc config, the changes in it will be also reflected in its target documents and so on.

Item's format

The format of a DocUpDoc config item:

D2D[name] D4D[name] NEW{ condition } DEL[subtags] DELAY{ condition }

The format description:

D2D[name] - the name of a config DocToDoc. If a change described by an item of the DocToDoc config occurs in the source document, the change will be reflected in the target documents;

D4D[name] - the full name of a DocForDoc config's item to get the target documents for their update. You can type here the extended form of the documents getting: **D4D[SRC{ condition } name TRG{ condition }]** - read more about the format in the end of the help topic - DocExt\Doc config;

NEW or **NEW{ condition }** - means that if a new source document is being saved for the first time, the data from it will be added into the target documents. If the tag is typed with a condition, it will be checked on each target document;

DEL[subtags] - the part of an config item defines the action on the target documents when the source document is being deleted. The tag contains the subtags inside:

FAIL or **FAIL{ condition }** - means the deletion fail if the result of the condition's check on a target document of any level in the updates chain is @True (1). Type the tag without a condition in order that just presence of target documents causes the deletion fail;

STOP or **STOP{ condition }** - works like FAIL, but is checked the target documents only of the first level in the updates chain;

ERASE or **ERASE{ condition }** - defines when values of the source document will be erased from the target documents;

ALSO or **ALSO{ condition }** - defines when the target documents will be also deleted together with the source document.

If you want to perform a subtag action unconditionally, just type the tag without the brackets { }. The subtags at source document deletion will be applied in the same sequence as they is listed in the description of the tag DEL until the first check condition result of a subtag @True (1).

DELAY or **DELAY{ condition }** - defines when the target documents will be updated through server. It can be useful in case the user who change the source document has not access to perform the updating by the config item DocUpDoc in target documents. Type only the tag DELAY without the brackets { } to perform the delayed updating unconditionally.

2.3 DocToDoc type

Description

A DocToDoc config defines the part of data which will be transferred from a source document to a target one. Through the config you can manage the rules when and what data will be transferred between the two documents. Moreover, you can add the post-calculation for a target document that will be performed just after the transferring. The config also supports the multiple transfer multi-to-one that means the values of source document is being placed to values list of target one.

A DocToDoc config consists from its items. Type each item on new line. A config's item can define the data for transfer (transfer's item) or call other DocToDoc config (link's item). The items of both types can be typed in a config DocToDoc at the same time.

You can address to a config DocToDoc from a config item DocUpDoc and some public functions of the library DocExtLib.

Adding

To create a config DocToDoc perform:

- Open the view **Configs \ DocToDoc** and in its actions bar press **Create DocToDoc Config**;
- On the opened form in the field **Name** type the config name without spaces. It must be unique name among all DocToDoc configs. Using the name you can address to the config from a code and from other configs;
- At the right define whether the config is enabled or not. To a config not marked as enabled you can not address at all;
- In the field **Description** type text about the config. For example - From Company to Contract. It's help info that will not be used anywhere;
- In the fields **Version** set the number and date of the config version. It's help info that will not be used anywhere.

Item's format

An item can consist from prefix part, main part and target's part. Only main part is always required.

[Prefix part] Main part [Target part]

Prefix part

The prefix part defines whether the transfer data are the defaults or the transfer is multi-to-one or adds an other DocToDoc config to the current. It's defined by the tags:

DFT - means the transfer data are defaults and not will be updated after the transferring if the source document is changed or deleted. It relates to updating through a DocUpDoc config where the DocToDoc config is used. In a code you can control the transfer by the item through input value of a used function.

MLT - means that value of source document is placed in values list of target document (the multi-to-one transfer). Use the tag if you want to place values from more than one source document into one target document. You can type the tag in more than one config item.

KEY - matters only if the tag MLT in the config item is applied. Means that the source value is key value among all values transferring to the values list of target. All together the transfer items in a config with the tag KEY must provide the uniqueness for the values transferred by the items into target document at adding or updating. Otherwise the DocToDoc config will not be processed and the message box about the error will be displayed.

D2D - means an other DocToDoc config will be added to the current config. If you use the DFT, MLT, KEY it will be applied to all items of the added DocToDoc config. In the added config you also can use the tag D2D to add an other config and so on.

Main part

The main part of a config's item defines what data of source document (source for short) and to what field of target document (target for short) will be transferred. Use one of possible format, when typing the main part of a DocToDoc config item:

name - means the name of identical fields of source and target or the name of an other DocToDoc config if the tag D2D is typed. Just type the name of a field or a config.

SRC{ formula } - means the computing the name of identical fields of source and target or the name of an other DocToDoc config if the tag D2D is typed. Type a formula that will be computed on source to get the name of a field or a config.

TRG{ formula } - means the computing the name of identical fields of source and target or the name of an other DocToDoc config if the tag D2D is typed. Type a formula that will be computed on target to get the name of a field or a config.

BOTH{ formula } - means the computing the name of identical fields of source and target or the name of an other DocToDoc config if the tag D2D is typed. Type a formula that will be computed the both on source and on target at the same time to get the name of a field or a config. Inside the formula type the tags SRC[name] and TRG[name] to use values of the fields with the specified names in source and in target accordingly. A result got by a tag SRC or TRG will always be a text enclosed in quotes. If the result will be a list, each its element will be enclosed in quotes with separating the list elements by colon between themselves to represent them in the format of text list in Formula Language.

All tags described below can be used only if the tag D2D was not typed in the config item.

name1 or **{ formula } >> name2** - means the fields' names of source and target are not identical or instead a field's name of source is used an formula. Type a field's name or a formula (in the brackets) of source at the left of ">>" and a field's name of target at the right.

name or **{ formula } >> SRC{ formula }** - means the field's name of target will be computed on source. Type a field's name or a formula (in the brackets) of source at the left of ">>" and an formula in the tag SRC{ } at the right that will be computed on source to get the field's name of target.

name or **{ formula } >> TRG{ formula } -** means the field's name of target will be computed on target itself. Type a field's name or a formula (in the brackets) of source at the left of ">>" and an formula in the tag TRG{ } at the right that will be computed on target to get its field's name.

name or **{ formula } >> BOTH{ formula } -** means the field's name of target will be computed both on source and on target at the same time. Type a field's name or a formula (in the brackets) of source at the left of ">>" and a formula in the tag BOTH{ } at the right that will be computed both on source and on target at the same time to get the field's name of the target. How is working the tag BOTH{ } is described a bit above.

In all cases when a formula is used to compute the name of a target's field or an other config DocToDoc, the empty text as result means that the config item will not be applied.

Target part

After the main part of DocToDoc item you can type the target's part in the following format:

D4D[name] - means that instead of the changed value of the source document will be used values list from the documents got for the target document by the config item DocForDoc with the specified name. You also can type the tag D4D{ formula } to compute the name of a DocForDoc config item on target document. If the source document is not new it must be among the got documents, otherwise it cause the error. You can leave the got list of values as result or process it with tag { formula } typed next. If the tag D2D has been typed in the prefix part of a config item, the tag D4D to all items of the added DocToDoc config will be applied.

{ formula } - means the formula in the brackets will be computed on target and the result becomes the value its field. Matters only for an item without the tag D2D. The computation occurs only if the value of source defined in the config item was changed. The transfer of value of source to field of target before the formula computing will be done only at presence of the name of the field in the formula (case sensitive).

FAIL{ condition } - defines when the result of the transfer causes the fail. Type in the formula the name of field changed by the item as well as other fields names of target if it is required. Matters only for an item without the tag D2D. The formula result @True (1) or any not empty text will cause the error with message box and stops the transfer by the config. If the DocToDoc config is called from a DocUpDoc item of DocExt config before a source document is being saved the update process of related target documents will be stopped and the method Update (DocExt class) returns False (0).

2.4 Project type

Description

A Project config defines app or group of apps united by any sense (for example - by year, by project's name or by both) - further the project-forming apps. Besides, a Project config must contain the links to all apps to which an addressing is possible from the project-forming apps - further the project-related apps.

The first item of a Project config defines the defaults and each next item defines a project-forming app or refers to a project-related app in an other Project config. Use new line for each item at the config typing.

To the app described by a Project config item you can be addressed from a DocForDoc config item and some public functions of the library DocExtLib.

Adding

To create a config Project perform:

- Open the view **Configs \ Project** and press **Create Project Config** in its actions bar;
- On the opened form in the field **Name** type a name for the config without spaces. It must be unique name among all Project configs. Using the name you can address to the config from a code or from other configs;
- At the right define whether the config is enabled or not. To a config not marked as enabled you can not address at all;
- In the field **Description** type a text about the config. For example - Contracts 2015. The description can be shown in some message boxes at code running;
- In the fields **Version** set the number and date of the config version. It's help info that will not be used anywhere.

Item's format

Use the first item in a Project config to set defaults for project-forming apps in the config. If you don't type the item with defaults separately, they will be defined by the first item with an app in the config.

The format of config's item with defaults:

SRV[name] FOLDER[name]

The format description:

SRV[name] - the abbreviated name of the default server;

FOLDER[name] - the name of a default file folder relative to data folder of Notes client or Domino server.

For a config's item that defines a project-forming app use the format:

name [description] SRV[name] FILE[name] RID[identifier]

The format description:

name - the unique name for an app in the config. Using the name you can address to the app, that is being described in the config item, from other configs or a code.

[description] - the description of the app in the project. The description can be displayed in a message box at a code running.

SRV[name] - the abbreviated name of the server with the app. Omit the tag to use the default server's name specified in the first config item.

FILE[name] - the name of the file of the app. Type full file name relative to Notes client / Domino server data folder or type the symbol "\" at the beginning to type the file name relative to the default folder specified in the first config item.

RID[identifier] - the eight-bit replica ID of the app. Type the tag for each app or instead press the button "Set Replica ID" at the bottom before the config saving to add the tag RID with the replica ID for all apps in the config where the tag RID wasn't typed.

3 Library DocExtLib

Description

The DocExtLib is alone Lotus Script library and it should be pasted into all apps which you plan to use the Doc Ext Configs (DEC) in.

The library works with config documents defined in the app DEC by an admin.

The library contains the main class DocExt, the code of its processing and few useful public functions that can be called from the apps which the library DocExtLib has been previously pasted into.

3.1 DocExt class

Description

The class DocExt is the main public class of the library DocExtLib. An object of the class is indissolubly related to a document of an app. Just give the document when you are creating DocExt object. An object of the class is working with the config document of DocExt type named as the form of the document which the object is related to.

Object creating

You need create a DocExt object before the document will be changed. Thus for a UI document changing by a user through its form you should type the DocExt object's creating in the event PostOpen of a form. For a back-end document type the DocExt object's creating for it before all changes of the document in a code.

The object dimension:

dim doc as NotesDocument

dim ext as DocExt

The simple creating:

set ext = new DocExt(doc, "")

The conditional creating:

```
if not DEdocLocked( doc ) then
        set ext = new DocExt( doc, "" )
end if
```

At the conditional creating of an object (*ext*) the document (*doc*) is checked for its locking by the config DocExt\Doc. If the document isn't locked, the function DEdocLocked returns the result False (0).

The typical code which you should type in the event PostOpen of a form:

```
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
```

Object processing

The processing of a DocExt object performs all validations provided by its config DocExt\Doc on the object's document, and then, if the validations have been passed successfully, updates all related documents with changes logging.

To call a DocExt object's processing in your script you must call the object's method named *Update*. The method should be called before the document will be saved. Thus for a UI document changing by a user through its form you should call the DocExt object's processing in the event QuerySave of the form. For a back-end document you should call the DocExt object's processing just before the document saving in a code.

The method *Update* returns a result: True (-1) or False (0). The result True means that all validations of the document were successfully passed and related documents have been updated. The validations are defined by the items Doc and the updates are defined by the items DocUpDoc of the config DocExt related to the document by the name of its form.

Type the code in form's event QuerySave to call the processing of a DocExt object (*ext*) at the UI document saving:

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

Type the code to call the processing of a DocExt object (*ext*) at the back-end document saving (doc):

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

3.2 Public functions

Description

In the library DocExtLib are few public functions that can be called from your code to process the documents in apps. Usually such function addresses to one or more config documents in the app DEC to perform the action of the function. All the public functions of the library are begun from two symbols - DE.

3.2.1 Function DEdocDescr

Gets a document description consisting from its type and computed name.

Syntax

DEdocDescr(doc, docTypeNum%)

Arguments

doc - NotesDocument, input. A document whose description should be got.

docTypeNum% - integer, input. Case number of type to be got. For some languages (for English does not matter).

Return

String.

Configs used

DocExt\Doc named as the document form.

3.2.2 Function DEdocID

Gets the ID of a document in the same format that is used as document tag in the documents list got as result of various public functions of the library DocExtLIb.

Syntax

DEdocID(doc)

Arguments

doc - NotesDocument, input. A document whose ID is getting.

Return

String.

Configs used

None.

3.2.3 Function DEdocLocked

Checks a document for its locking. The locking here means that a document can't be edited or deleted. The locking of a document is defined by the tags LOCK typed in DocExt\Doc config named as the document's form. You should call the function when creating an object of DocExt class for a document before its changing in your code. Read more about this in the help topic - Creating object of DocExt class.

Syntax

DEdocLocked(doc)

Arguments

doc - NotesDocument, input. A document whose locking should be checked.

Return

Boolean. If the document is locked, the function returns the result True (-1). Otherwise the result is False (0).

Configs used

DocExt\Doc named as the document form.

3.2.4 Function DEgetApp

Gets an application.

Syntax

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

Arguments

prjName$ - string, input. The name of the Project config where the app is described. The empty string means the Project config where the current app which the function is called from is described.

appName$ - string, input. The name of the app in the Project config.

silent - boolean, input. The value True (-1) means that if the app hasn't been got, this will not cause the displaying of message box with the error.

db - NotesDatabase, output. The app that should be got.

Return

Boolean. The result True (-1) means that the app has been got and False (0) that not.

Configs used

Project.

3.2.5 Function DEgetAppName

Gets the name of an application from the Project config which it has been described in.

Syntax

DEgetAppName(db)

Arguments

db - NotesDatabase, input. An app whose name should be got. If was given Nothing, the name of the current app which the function is called from will be got.

Return

String. If the Project config with the app description has not been got, the function returns empty string.

Configs used

Project.

3.2.6 Function DEgetC4D

Gets the target documents for a source document.

Syntax

DEgetC4D(D4Dname$, trgCond$, srcDoc, silent, trgDocs)

Arguments

D4Dname$ - string, input. The name of a DocForDoc config's item which the target documents should be got by. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items you should type their names in the string through a space.

trgCond$ - string, input. An additional condition in Formula Language to select the target documents got by the DocForDoc config's item.

srcDoc - NotesDocument, input. A source document which the target documents should be got for.

silent - boolean, input. The value True (-1) means that if the target documents have not been got, no message box about this will be displayed.

trgDocs - NotesDocument List, output. The target documents that should be got. The list can contain documents from different apps.

Return

Boolean. The result True (-1) if at least one target document has been got and False (0) if not.

Configs used

DocExt\DocForDoc.

3.2.7 Function DEgetD4D

Get the only target document for a source document.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocForDoc config's item which the only target document should be got by. Read more about the name in the help topic - DocExt\DocForDoc config.

trgCond$ - string, input. An additional condition in Formula Language to select the only target document from got ones by the DocForDoc config's item.

srcDoc - NotesDocument, input. A source document which the only target document should be got for.

silent - boolean, input. The value True (-1) means that if exactly one target document hasn't been got, no message box about this will be displayed.

trgDoc - NotesDocument, output. The only target document that should be got.

Return

Boolean. The result True (-1) means that exactly one target document has been got and False (0) if no one or more than one.

Configs used

DocExt\DocForDoc.

3.2.8 Function DEgetPrjName

Gets the name of Project config which a given application is described in.

Syntax

DEgetPrjName(db)

Arguments

db - NotesDatabase, input. An app which the name of Project config should be got for. If you give Nothing, the name of Project config with description of the current app which the function is called from will be got.

Return

String. If the Project config with description of the app has not been found, the function will return empty string.

Configs used

Project.

3.2.9 Function DEgetSrcC4D

Gets documents (further data sources) for a source document and transfers data from them into a target document. Combines the consequent calls of two functions - DEgetC4D and DEprD2D.

Syntax

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

Arguments

D4Dname$ - string, input. The name of DocForDoc config's item which the documents should be got by. Read more about this name the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names in the string through a space.

D4DtrgCond$ - string, input. An additional condition in Formula Language to select the documents got by the DocForDoc config's item.

D4DsrcDoc - NotesDocument, input. A source document which the data sources should be got for.

D4Dsilent - boolean, input. The value is True (-1) means that if the data sources have not been got, no message box about this will be displayed.

D2Dname$ - string, input. The name of DocToDoc config by which the data from the got data sources should be transferred into a target document. To use several configs at the same time you should type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - to add data, 3 - to erase data. Read more about this number in the help topic - Function DEprD2D.

D2DtrgDoc - NotesDocument, modified. The target document which the data from the got data sources should be transferred into. The function can change the document but doesn't save it.

Return

Boolean. The result True (1) means that the data sources were got and some their data have been transferred into the target document and False (0) that at least one of this hasn't occurred.

Configs used

DocExt\DocForDoc, DocToDoc.

3.2.10 Function DEgetSrcD4D

Gets the only document (further data source) for a source document and transfers data from it into a target document. Use the function instead of consequent calls of two other functions - DEgetD4D and DEprD2D, to reduce count of code lines and exclude presence of data source in them.

Syntax

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

Arguments

D4Dname$ - string, input. The name of DocExt\DocForDoc config's item which the data source should be got by. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time, type their names in the string through a space.

D4DtrgCond$ - string, input. An additional condition in Formula Language to select the data source from got ones using the DocExt\DocForDoc config's item.

D4DsrcDoc - NotesDocument, input. The source document which the data source should be got for.

D4Dsilent - boolean, input. The value is True (-1) means that if the data source has not been got, no message box will be displayed.

D2Dname$ - string, input. The name of DocToDoc config by which the data from the got data source should be transferred into the target document. To use several configs at the same time, type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - to add data, 3 - to erase data. Other values are also provided. Read more about this number in the help topic - Function DEprD2D.

D2DtrgDoc - NotesDocument, modified. The target document which the data from the got data source should be transferred into. The function can change the document but doesn't save it.

Return

Boolean. The result True (1) means that the data source was got and some its data have been transferred into the target document, and False (0) that at least one of this hasn't occurred.

Configs used

DocExt\DocForDoc, DocToDoc.

3.2.11 Function DEnotGetC4D

Checks for absence of target documents for a source document. Use the function instead of the function DEgetC4D if you don't want to have the target documents in your code but want only to check for their presence or absence.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item which the absence of target documents should be checked by. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time, type their names in the string through a space.

trgCond$ - string, input. An additional condition in Formula Language to select target documents from got ones using the DocExt\DocForDoc config's item.

srcDoc - NotesDocument, input. A source document which the absence of target documents should be checked for.

silent - boolean. The value True (-1) means that if the target documents were got, no message box about this will be displayed.

Return

Boolean. The result True (-1) if no target documents were got and False (0) if were.

Configs used

DocExt\DocForDoc.

3.2.12 Function DEopenD4D

Opens a target document got for a source document. If several target documents have been got, the view for choice what of them should be opened will be appeared.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item which the target document should be got by. Read more about this name in help topic - DocExt\DocForDoc config. To use several configs names at the same time, type their names in the string through a space.

trgCond$ - string, input. An additional condition in Formula Language to select the target document from got ones using the DocExt\DocForDoc config's item.

srcDoc - NotesDocument, input. A source document which the target document should be got for, to open it next.

silent - boolean, input. The value True (-1) means that if no documents haven't been got for the opening an empty view will not be displayed.

Return

Boolean. The result True (-1) means that documents have been got and at least one of them has been opened by a user.

Configs used

DocExt\DocForDoc.

3.2.13 Function DEpickC4D

Gets the target documents for a source document in order that a user next picks some of them. Differs from the function DEpickD4D that more than one document can be picked.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item by which the target documents should be got in order that a user next picks some of them. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names in the string through a space.

trgCond$ - string, input. An additional condition in Formula Language to select the target documents from got ones using the DocExt\DocForDoc config's item.

srcDoc - NotesDocument, input. A source document for which the target documents should be got.

showEmpty - boolean, input. The value True (-1) means that if targets documents have not been got, the view with no documents for picking will not be displayed.

prompt$ - string, input. A text that should be displayed at the documents picking. For example - describe why the picking is being done.

trgDocs - NotesDocument List, modified. If the target documents have been picked by a user, they will be placed into the list. Otherwise the list will remain with no changes.

Return

Boolean. The value True (-1) means that the target documents were got and next a user has picked some of them, and False (0) that at least one of this hasn't occurred.

Configs used

DocExt\DocForDoc.

3.2.14 Function DEpickD4D

Gets the target documents for a source document in order that a user next picks the only one of them. Differs from the function DEpickC4D that only one document can be picked.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item by which the target documents should be got in order that a user next picks the only one of them. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names through a space.

trgCond$ - string, input. An additional condition in Formula Language to select the target documents from got ones using the DocExt\DocForDoc config's item.

srcDoc - NotesDocument, input. A source document which the targets documents should be got for.

showEmpty - boolean, input. The value True (-1) means that if targets documents have not been got, the view with no documents for picking will not be displayed.

prompt$ - string, input. A text that should be displayed at the document picking. For example - describe why the picking is being done.

trgDoc - NotesDocument, modified. If a target document has been picked by a user, it will be set for the variable. Otherwise the variable will remain with no changes.

Return

Boolean. The value True (-1) means that the target documents were got and a user has picked the only one of them, and False (0) that at least one of this has not occurred.

Configs used

DocExt\DocForDoc.

3.2.15 Function DEpickDocs

Allows a user to pick the documents of granted ones for a source document. Differs from the functions DEpickC4D and DEpickD4D that the documents for the picking are granted into the function but don't get by using it.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item with which the granted documents should be picked by a user. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names through a space.

srcDoc - NotesDocument, input. A source document for the DocExt\DocForDoc config's item.

showEmpty - boolean, input. The value True (-1) means that if even no documents were granted for the picking, the view will be also displayed.

multi - boolean, input. The value True (-1) means that more than one document can be picked by a user.

prompt$ - string, input. A text which should be displayed at the picking. For example - describe why the picking is being done.

trgDocs - NotesDocument List, modified. Into the function is given the documents list which a user should make a choice of. If a user has picked documents from the list, the list of the granted documents will be replaced with the list of picked ones.

Return

Boolean. The value True (-1) means that at least one document has been picked by a user and False (-1) if not.

Configs used

DocExt\DocForDoc.

3.2.16 Function DEpickSrcC4D

Gets the documents for a source document in order that a user picks some of them and transfers the data from the picked documents (further data sources) into a target document. Differs from the function DEpickSrcD4D that more than one document can be picked.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item by which the documents for picking should be got. Next a user picks some of them as data sources. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names in the string through a space.

D4DtrgCond$ - string, input. An additional condition in Formula Language to select documents from got ones using the DocExt\DocForDoc config's item.

D4DsrcDoc - NotesDocument, input. A source document which the picking documents should be got for.

showEmpty - boolean, input. The value True (-1) means that the view with no choices will be displayed even if the documents for picking have not been got.

prompt$ - string, input. A text which should be displayed at the documents picking. For example describe why the picking is being done.

D2Dname$ - string, input. The name of a DocToDoc config by which the data from the picked documents should be transferred into a target document. To use several configs at the same time you should type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - adding data, 3 - erasing data. Other values are also provided. Read more about this number in the help topic - Function DEprD2D.

D2DtrgDoc - NotesDocument, modified. The target document which the data from the picked documents should be transferred into. The function can change the document but doesn't save it.

Return

Boolean. The value True (-1) means that the documents were picked by a user and some data from them have been transferred into the target document and False (0) that at least one of this hasn't occurred.

Configs used

DocExt\DocForDoc, DocToDoc.

3.2.17 Function DEpickSrcD4D

Gets the documents for a source document in order that a user picks the only one of them and transfers the data from the picked document (further data source) to a target document. Differs from the function DEpickSrcC4D that the only one document can be picked.

Syntax

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

Arguments

D4Dname$ - string, input. The name of a DocExt\DocForDoc config's item by which the documents should be got for the picking. Next a user picks the only one of them as the data source. Read more about this name in the help topic - DocExt\DocForDoc config. To use several configs items at the same time you should type their names through a space.

D4DtrgCond$ - string, input. An additional condition in Formula Language to select documents from got ones using the DocExt\DocForDoc config's item.

D4DsrcDoc - NotesDocument, input. A source document which the picking documents should be got for.

showEmpty - boolean, input. The value True (-1) means that the view with no choices will be displayed even if the documents for picking have not been got.

prompt$ - string, input. A text which should be displayed at the document picking. For example - describe why the picking is being done.

D2Dname$ - string, input. The name of a DocToDoc config by which the data from the picked document should be transferred into the target document. To use several configs at the same time you should type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - adding data, 3 - erasing data. Other values are also provided. Read more about this number in the help topic - Function DEprD2D.

D2DtrgDoc - NotesDocument, modified. A target document which the data from the picked document should be transferred into. The function can change the document but doesn't save it.

Return

Boolean. The value True (-1) means that a document was picked by a user and some data from it have been transferred into the target document and False (0) that at least one of this has not occurred.

Configs used

DocExt\DocForDoc, DocToDoc.

3.2.18 Function DEprCopyDelay

Performs the delayed copying of a document into an application using a server. The function is useful if an user hasn't enough access level to an app but a document has to be added into it. The request for document's copying will be placed in the app DEC or an other app which is set to process the requests on server. Read more about this in the help topic - Delayed documents update.

Syntax

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

Arguments

doc - NotesDocument, input. A source document which should be copied. It can be a new or saved document.

dstDb - NotesDatabase, input. An application which the document should be copied into.

saved - boolean, input. The value True (-1) means that a saved document is being copied and the source document itself will be copied at result of the processing of the request for copying. The value False (0) means that a not saved is being copied and in this case all fields of the source document will be placed into the request for copying.

Return

Does not return a value.

Configs used

None.

3.2.19 Function DEprD2D

Transfers the data from a source document to a target document.

Syntax

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

Arguments

D2Dname$ - string, input. The name of a DocToDoc config which the data should be transferred by. To use several configs at the same time you should type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - adding data, 3 - erasing data.

You also can use extended numbers instead of 1 and 3 for special data transfer:

11 - adding data without defaults in a DocToDoc config (items with tag DFT). Just the number 1 adds data with defaults.

12 - adding data even if the values of source document (items with tag MLT) can not be added because they already are in the values list of target document. Just the number 1 will cause error in this case.

13 - works like number 12 but if the values of source document (items with tag MLT) were found in the values list of target document, all defaults in the values (items with both tags MLT and DFT) will be updated.

31 - erasing data with defaults (items with tag DFT). Just the number 3 erases data without defaults.

32 - erasing data even the values of source document (items with tag MLT) were not found in the values list of target document, the whole list with values of target document will be displayed in order that an user picks what position in the list should be erased.

srcDoc - NotesDocument, input. The source document which the data should be transferred from. It can be Nothing only for the data erasing if the DocToDoc config hasn't items computing on a source document (tag SRC).

trgDoc - NotesDocument, modified. The target document which the data should be transferred into. The function can change the document but doesn't save it.

Return

Boolean. The value True (-1) means that some data from the source document were transferred into the target one.

Configs used

DocToDoc.

3.2.20 Function DEprD2Ddelay

Performs the delayed transferring the data from a source document into a target document using a server. The function can be useful if an user hasn't enough access level to change a document but some data have to be transferred into it. The request for document update will be placed in the app DEC or other app which is set to process the requests on server. Read more about this in the help topic - Delayed Documents Update.

Syntax

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

Arguments

D2Dname$ - string, input. The name of a DocToDoc config which the data should be transferred by. To use several configs at the same time you should type their names in the string through a space.

D2Dprocess% - integer, input. The number of a transfer process: 1 - adding data, 3 - erasing data. Read more about this number in the help topic - Function DEprD2D.

srcDoc - NotesDocument, input. The source document which the data should be transferred from. It can be Nothing only for the data erasing if the DocToDoc config hasn't items computing on a source document (tag SRC).

trgDoc - NotesDocument, modified. The target document which the data should be transferred into. The document can be changed using the request which will be sent onto server for its processing. You must not next save the document in your code to save the changes in it because this will be done at processing of the request on server by the special agent.

Return

Boolean. The value True (-1) means that some data from source document will be transferred into target document using the request for delayed document update.

Configs used

DocToDoc.

3.2.21 Function DEprDelay

Processes a request for delayed document update using a server. The function is used in the special agent and in the view action. The agent should be run on server in the app DEC or in an other app which is used instead of it. The action you should use to understand any problem at the code debugging.

Syntax

DEprDelay(delayDoc)

Arguments

delayDoc - NotesDocument, modified. The document represents a request for delayed document update in an app. Read more about this document in the help topic - Delayed Documents Update.

Return

Does not return a value.

Configs used

DocExt\DocUpDoc named as form of the modified document.

3.2.22 Function DEprDelete

Deletes a document if it is not locked by the config DocExt\Doc with deleting or updating all related documents and with the logging these actions. The related documents are defined by the config DocExt\DocUpDoc (items with tag DEL) named as the form of the deleted document. You should use the function in the sub QueryDocumentDelete of the Database Script in your apps.

Syntax

DEprDelete(doc)

Arguments

doc - NotesDocument, modified. The document which is being deleted.

Return

Boolean. The value True (-1) means that the document wasn't locked and was successfully deleted as well as all related documents were successfully deleted or updated.

Configs used

DocExt\Doc, DocExt\DocUpDoc.

3.2.23 Function DEprMove

Moves a document and all related ones into a project with the logging. The documents are moved to the apps of an other project. Each moving document will be moved to the destination project's app which is named just as the current app of the document is named in the source project. The related documents are defined by the config DocExt\DocUpDoc (items with the tag DEL[ALSO]) named as the form of the moved document.

Syntax

DEprMove(doc, trgPrjName$)

Arguments

doc - NotesDocument, modified. The document which should be moved.

trgPrjName$ - string, input. The name of a Project config with the app which the document should be moved into.

Return

Boolean. The value True (-1) means that the document itself and all related ones were successfully moved to the apps of the destination project.

Configs used

Project, DocExt\DocUpDoc.

3.2.24 Function DEpromptApp

Allows a user to pick an application from a Project config.

Syntax

DEpromptApp(prjName$, appName$, appDescr$)

Arguments

prjName$ - string, input. The name of a Project config which an app should be picked from.

appName$ - string, output. The name of the picked app in the Project config.

appDescr$ - string, output. The description of the picked app. It's a text in the brackets [ ] following the app's name in the Project config's item.

Return

Boolean. The value True (-1) means that an app has been picked by a user.

Configs used

Project.

4 Settings

4.1 Start of the App using

App placement

For a company it's enough to use the only app DEC which the configs for all documents types from all apps of the company will be placed in.

For locating the app DEC is used the NOTES.INI variables: DE_CONFIG_SERVER - server where the app is placed, DE_CONFIG_FILE - file name relatively of Notes/Domino data folder or replica ID of the app.

Initially is supposed that the app will be placed into the data folder of apps' server with the file name DEconfig.nsf. If it was done any other actions are not required. The NOTES.INI variables on a user's computer will be set for using this app at the first access to it.

If you want to place the app in an own folder you must set the NOTES.INI variables. You can do it using the Notes menu item: Actions \ Doc Ext Configs \ Set this App for this computer.

Access level to App

It's enough to set access level to the app "Reader" for all users in order that they can address to the configs at working with apps. You can restrict access to a special config using the fields Readers and Authors in the config document.

The two roles are provided in the app: [ReadAll] - always is placed in the Readers field and allows to read any config, [EditAll] - always is placed in the Authors field and allows to edit any config.

4.2 Documents Update Log

The app DEC allows to log the changes of documents as well as their creation and deletion. To log a document creating and updating, you must use for it the object of DocExt class (script library DocExtLib) in a code. To log a document deletion you must paste special code into the event QueryDocumentDelete of the app section Database Script. The document fields whose changes should be logged are defined by DocExt\Doc config named as the form of changed document.

App using

The app DEC contains the design elements to deploy the documents update log. You can use the app DEC itself as the log but it's more convenient to use instead of it a stand-alone app with some its design elements. You also can use the same app for the processing of delayed documents update.

You should describe in a Project config the app chosen as the documents update log. It can be a config where also are described the apps with documents whose updates should be logged, or can be a separate config with the log which the links from other Project configs have been typed to. It's defined by how many logs you want to use. By default the app's name in a Project config by which you will address to log is defined by the variable DE_LOG_APP of the script library DocExtLib with initial value "Log". You can overwrite the default app's name using the tag LOG_APP in a DocExt\Doc config.

Design elements

In the list are specified the design elements in the app DEC that are required for deploying the documents' update log. If you have decided to use stand-alone app as the log, you should create a new app and paste the design elements into it.

- The view "Documents Update Log" - for viewing the documents updates.
- The form "Documents Update Log - Entry" - displays the update details of a document.

Log entry

A log entry is a document in log which can be displayed using the form "Document Update Log". Such document contains the fields about a change of a document itself as well as about all updates in other documents related to the change:

- DocFile - the file name of the app with the changed document;
- DocServer - the server name of the app with the changed document;
- DocUNID - the UNID of the changed document;
- Message - text list about the document change. The first item in the list contains info about the user who did the changes, and about the changed document. And each next item describes the change of a document field;
- Targets - Rich Text about the updates in all related documents that were caused by the change. Besides text the field contains links to updated related documents;
- Time - the time when the changes were done;
- User - the user name who did the changes.

Log into document field

Besides the documents' updates logging into app you also can log them into a special field of changed document. By default the field name is defined by the variable DE_LOG_FIELD of the script library DocExtLib with the initial value "Log". You can either change value of the variable value or set a value that differs from the default by the tag LOG_FIELD in a DocExt\Doc config. The field must be placed on the document form with the option "Allow multiple values" and the values separator "New Line" in order to correctly display the document updates log.

4.3 Delayed Documents Update

The app DEC allows to deploy the delayed documents update through a server. The delaying can be defined by a Doc\DocUpDoc config and some special functions of the script library DocExtLib. At the saving of a document and starting the updating of all related documents, if an user hasn't access to edit some of the documents, but the delayed update is enabled, the documents will be updated through the sending the requests onto the server. By default the requests for documents update are sent to the special mail-in database on the server where they will be instantly processed by the special agent.

App using

The app DEC contains the design elements to deploy the delayed documents update. For the deploying you can use the app DEC itself but is more convenient to use instead of it a stand-alone app with some its design elements. You can also use the same stand-alone app to deploy documents' update log.

Desing elements

In the list are specified the design elements of the app DEC that are required for deploying the delayed documents update. You should copy them to new stand-alone app if you have decided to use it for this purpose instead of DEC.

- The script library DocExtLib. It will process the changes delays;
- The agent "Delayed Documents Update - Processing Incoming Request", trigger - "Before new mail arrived". It will instantly process a request for document update arrived as e-mail message.
- The agent "(Delayed Documents Update - Process Requests)". It's running on selected in view documents. You should use the agent for manually processing a request for document update to get info about an error because of which it was not processed automatically.
- The agent "(Delayed Documents Update - Set Requests as Unprocessed)". It's running on selected in view documents. The agent sets the status New in a request for document update that it could be processed again.
- The view "Delayed Documents Update". For viewing the requests for documents update and troubleshooting.
- The form "Delayed Documents Update - Request". The form displays the request for a document update.

Mail-in database

To use the delayed documents update you must set mail-in database on server in order to receive the requests for the documents update. By default is expected the using of the address DEconfig. If you want to use an alternate address type it as value of the variable DE_DELAY_MAIL in the app DEC. Define what app you will use to deploy the delayed documents update and set it as mail-in database on server. Next make sure that the agent "Delayed Documents Update - Processing Incoming Request" is enabled in the app.

Request for document update

A request for document update is document that is being sent to the special app chosen to process the requests (mail-in database). A request contains full info about the document update including the service fields and the all fields which should be updated in the target document. A request can be viewed through the form "Delayed Documents Update - Request" (DocUpDelay) in the view "Delayed Documents Update" of the app processing the requests.

The service fields of a update request are begun on DEdelay:

- DEdelay - the sign of that the document is request;
- DEdelayError - the existence of the field means the request was processed. Any text in the field means the request was processed with an error and describes it;
- DEdelayTime - the time when the request was created;
- DEdelayTrgDoc - the UNID of target document that should be updated;
- DEdelayTrgFields - the names of fields of target document. All the fields listed here are present in the request with values that should be set in the target document;
- DEdelayTrgFile - the file name of app with target document;
- DEdelayTrgServer - the server name of the app with target document.

Variables NOTES.INI on server

Make sure that the variables DE_CONFIG_SERVER, DE_CONFIG_FILE, DE_CONFIG_LANGUAGE are set in NOTES.INI file on server which the app for processing the requests for documents update is placed on. Read more about these variables in the help topics: Start of the App Using and Script Library Variables**.**

4.4 Script Library Variables

In the app DEC are used the variables of the script library DocExtLib to set some parameters of app working and to translate into a required language all app messages that are displayed at the code running. The values for the variables can be redefined through special profiles in the app. The value setting for a variable must satisfy to the Lotus Script syntax, otherwise it causes an error and the profile will not be saved. Two types of profiles are provided in the app: Common and Language.

Common profile

The profile contains the variables which affect different aspects of working the app DEC. The single common profile in the app sets values for the following variables:

- DE_DEBUG_D2D - boolean, initially False. Sets the displaying of all field changes caused by a config DocToDoc. The value True enables such displaying, otherwise, if value is False, only changes of fields listed in the config DocExt\Doc matching the modifiable document are being displayed.
- DE_DELAY_MAIL - string, initially "DEconfig". The e-mail address of mail-in database which is set to receive and process the requests for delayed documents updates. If you type the empty string as the variable's value but the delayed documents update is defined in DocExt\DocUpDoc config, the created requests for documents update will be only saved in the used app DEC without their further processing.
- DE_LOG_APP - string, initially "Log". The name of the app (in a Project config) by default which is used as the documents updates log.
- DE_LOG_FIELD - string, initially "Log". The name of the field by default which is used to log document changes in the document itself.

- DE_PRINT - boolean, initially True. Sets the displaying of any messages at application working. The value False disables this displaying.
- DE_SHOW_PROGRESS - long, initially 10. The minimum count of related documents which are updated at the changing of source document at which the progress bar of the updating should be displayed. Set the zero (0) not to display the progress bar at all.

You can change a variable value through the Notes menu: Actions \ Script Library Variables \ Common - Edit.

Language profile

A language profile defines a language for messages that are being displayed at the code running of library DocExtLib. A profile contains the variables to translate all messages into a language. Initially the app DEC always contains English language profile. To add new language profile or to edit the variables of an existing one, choose in Notes menu: Actions \ Script Library Variables \ Language - Edit.

Setting a Language

If you have added own language profile you can set it as default for the app DEC through the Notes menu item - Actions \ Script Library Variables \ Language - Set default for app. By this you set the chosen language name as value for the field Language in the common profile and next the name will be propagated as value of the variable DE_CONFIG_LANGUAGE in the file NOTES.INI at the first access to the app DEC on a computer. To set an alternate language for a computer you must change the NOTES.INI variable's value to the language name which the profile had been created for. A one method to do this, choose in the Notes menu: Actions \ Script Library Variables \ Language - Set for this computer.