Formula(r).WorkFlow is the one and only „all-in-one device" suitable for any workflow application with Lotus Domino.

This document describes the creation of a simple workflow in a custom application with Formula(r).WorkFlow.

1 Copy standard design elements

Please copy the forms, views, agents, subforms and script-libraries mentioned in the table on the right handside in your application. Be aware to rename the form "(Config.Global.Target)"

Name	For code	For user	For Web
(Config.Global.Target) **fa_XXconfig**	X Rename to fa_config		
Views			
va_wfbycompleted va_wfbydate		X X	
va_wfbyinitiator va_wfbystatus		X X	
va_allsetup va_dspwfsteps	(X) X		
va_activities	X		
va_process	X		
va_myprocess	X		
va_wfautomation	X		
va_dspmineapproval	X		
va_wfoverdue	X		
Agents			
agentHandleOverdue	X		
agentHandleAutomation	(X)		
agentSaveWeb			X
Subforms			
sa_workflow sa_endworkflow	X X		
sa_workflow (web)			X
Script Libraries			
lib_WFFormula.VFL	X		
lib_WFInclude.VFL	X		

Copy the design elements above in your database

2 Configure the system

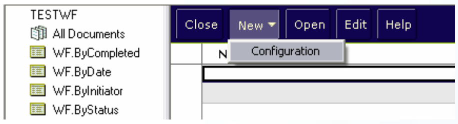
2.1 Your application database (Target database)

Open your database and go to the view "va_allsetup" and create a new "Configuration" document.
Here you can define:

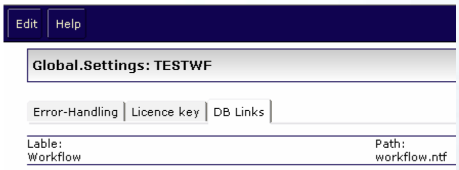
a) the error-handling (we use "MsgBox" for this example),
b) enter the licence key and also
c) enter the link to your WorkFlow configuration database containing all the roles and activity descriptions (in our example this is "**workflow.ntf**").

Be aware that the licence key for each application is different – so make sure you are using the right key !

Save and close the document afterwards.



Create a new configuration file in your database



Point the path to your configuration DB

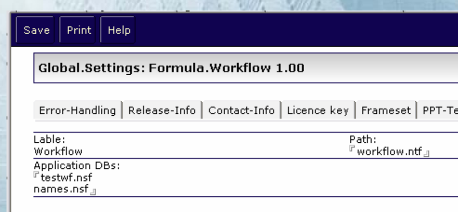
2.2 Workflow configuration database

Open the configuration database and go to the view "All documents" and open the "Config"-document.
There you should change the following 2 settings:

a) Licence key (otherwise you will receive prompts for the id time by time)
b) Location of your application database in the section "DB-Links – here you have to enter the path of your target database like "**wf\mydatabase.nsf**"

Save and close the document afterwards.

Configure the configuration database



Enter the link to the application databases (here « testwf.nsf » and « names.nsf »)

3 Define the process

In this example we will create a workflow for an "IT improvement request".
The user will create the request and submit it to the "Improvement manager". The IM will decide if it is an development or administration request and based on this judgement – either forward the request to any of these two departments or reject the request.

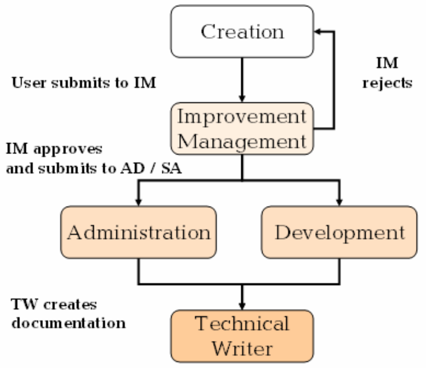
The responsible person in the dev- or admin department will then process the request and after completing all tasks.. forward this request to the IT documentation.
Here the technical writer will document this change request and finally close the request.

3.1 Flow chart

The flow chart diagram of the workflow can be seen on the right hand-side.
The following information can be derived from the chart:

a) we do have at least 5 different activities
b) we need to have at least 5 different activity owners
c) the process will split either in administration or development after approval of management

Now we are able to identify the activities in this process. For better readability in views we will add a numeric index to each activity to have the right sort order in views.



Flow chart of the process

3.2 Creation of access groups

First we will set up the access “groups” for this workflow.
Therefore we have to define the access rights from the specification first (see graphic on the right hand side):

Activity	Edit	Read
01 Creation	Initiator	
02 IM	CR_IM	Initiator, CR_IM
03 Admin	CR_Admin	Initiator, CR_Admin, CR_IM
03 Dev	CR_Dev	Initiator, CR_Dev, CR_IM
04 Technical Writer	CR_TW	Initiator, CR_IM, CR_TW

Access to the activities of the process

Also for easier identification of different roles or groups within a configuration database we suggest to add a “process identifier” as prefix to each access group.

Regarding the specification of the four existing elements (people, groups, roles and relations) it turns out that using “roles” might be the best opportunity for each of the activities.

3.2.1 01 Creation

Here we can use the standard role called “<Initiator>”.
This role gives author access to the creator (author) of the document by pointing to the internal field “fd_InitiatorVFL”. Because we will not add any other reader to this activity only the creator of the document is able to see and modify his request (and of course the “Admin”-role and the process owner that have always access).


Role: <Initiator> (1.00)

Name: <Initiator> Release: 1.00
Members: fd_InitiatorVFL Is this a field: yes
Surrogate: Max Muster/smartix/de Is this a field: yes
Email:
Description: Initiator of a document

Role pointing to the initiator and having a person as surrogate (01 Creation)

3.2.2 02 IM

The improvement manager is a person that has a name and will be identical for each request. Therefore we can use a role and select the person working on this activity from the N&A of our organisation.
Here we have defined an alternative email address to send all the notifications to “enteco-consult” instead of his Notes Address.
A solution like that might enable you to include other IT-systems to receive notifications about new requests.


Role: <CR_IM> (1.00)

Name: <CR_IM> Release: 1.00
Members: Mathias Bierl/smartix/de Is this a field: yes
Surrogate: Is this a field: yes
Email: mathias.bierl@enteco-consult.de
Description: The Improvement Manager of our organisation

Role of improvement manager with a person as responsible person (02 IM)

GMI KG · Gesellschaft für Migration und Integration
Bismarckstraße 81 · 70197 Stuttgart · Fon: +49 (0) 711 – 45 999 131
Fax: +49 (0) 711 – 45 999 135 · www.gesmi.de · info@gesmi.de

3.2.3 03 Admin

In this activity we will add all administrators of our organisation because we want the request to be processed as soon as possible. Therefore we use the organisation unit "Group" to add all existing administrators to this unit.
We can also define a surrogate like the name of the "manager" of our administration department as an escalation mechanism.

Groups: <CR_Admin> (1.00)			
Name:	<CR_Admin>	Release:	1.00
Manager:	Konrad Bachem/smartix/de		
Members:	Elke Kalka/smartix/de Eugen Heidebrecht/smartix/de	Surrogate:	
Mail-address:			
Description:	All Admins of our organisation		

Group « CR_Admin » to grant access for the admins (03 Admin)

3.2.4 03 Dev

Here we will do the same as for the admin activity and create another group for the development.
Unfortunately we do not have a separate group of people within our organisation – but we will define another manager ;-)

Alternatively again we can define another email-address for this group of people like a functional account or a mail-in database for documenting all the requests.

Groups: <CR_Dev> (1.00)			
Name:	<CR_Dev>	Release:	1.00
Manager:	Elke Kalka/smartix/de		
Members:	Eugen Heidebrecht/smartix/de	Surrogate:	
Mail-address:			
Description:	Group of developers in our organisation		

Group for the activity « 03 Dev »

3.2.5 04 Technical Writer

Here we assume that the technical writer for the administrative requests is different from the one for the development requests. Therefore we will first create 2 roles for the "TechWriter Admin" and "TechWriter Dev" and then use a relation to get the right role to our request.

For setting up this relation we will use the "relation" form and also use the enhanced formula language for evaluation this role:

GETROLE(Role_Name) ➔ This will add the role specified by „Role_Name"
GETGROUP(Group_Name) ➔ This will add the group specified by "Group_Name"

We will also use the standard field "fd_PreviousDecisionVFL" which contains the name of the previous activity.
Also we have to pass the GETROLE and GETGROUP-Elements as string as shown in the formula below:

```
@If(fd_PreviousDecisionVFL="03 Admin";
"GETROLE(<CR_TechWriter_Admin>)" ; "GETROLE(<CR_TechWriter_Dev>)")
```

Role: <CR_TechWriter_Admin> (1.00)			
Name:	<CR_TechWriter_Admin>	Release:	1.00
Members:	Kerstin Kluger/smartix/de	Is this a field:	yes
Surrogate:		Is this a field:	yes
Email:			
Description:	Technical writer for administrative changes		

Technical writer for administation (as role)

Role: <CR_TechWriter_Dev> (1.00)			
Name:	<CR_TechWriter_Dev>	Release:	1.00
Members:	Christian Henseler/EXT/smartix/de	Is this a field:	yes
Surrogate:		Is this a field:	yes
Email:			
Description:	Technical writer for development changes		

Technical writer for development (as role)

Relation: <CR_TW> (1.00)			
Name:	<CR_TW>	Release:	1.00
Type:	Formula		
Value:	@If(fd_PreviousDecisionVFL="03 Admin";"GETROLE(<CR_TechWriter_Admin>)";"GETROLE(<CR_TechWriter_Dev>)")		
Email:			
Description:	This role will check the previous activity name and then grant the right role the access to this activity		

Role for evaluating the routing depending

3.3 *Define the process*

Based on the flow chart on the side before we know the opportunities of routing for each activity.
First we have to create a “process-specification”.
Based on this document we can create activity specifications for each activity in the process.
Finally we can define the

a) access-rights,
b) notification settings and
c) routing options for each activity.

3.3.1 Creation of process definition

In the process definition document we have to enter the following values:

Name:	ChangeRequest
Release:	3.00
Owner:	People that can access and modify all documents of this process – Florian Lier/smartix/de
Reader:	People that can read each document – here the group of all Admins of smartiX
Job Name:	a @Formula to compute the job name
Description:	some text that enables you to understand this process even in a couple of years

Afterwards we are able to create all the activities by clicking on “Create activity”.

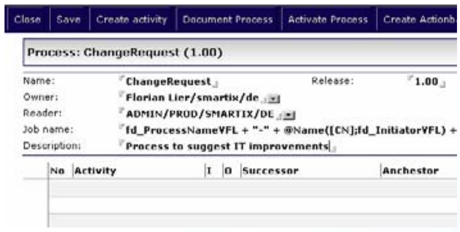
3.3.2 Creation of activities

For each activity we have to enter the following fields:

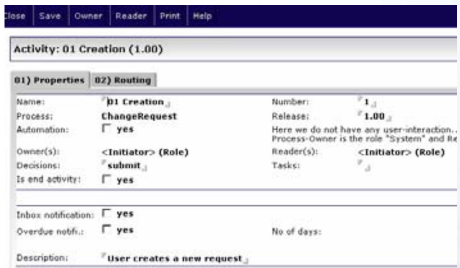
Name:	e. g. 01 Creation
Number:	sort order in the configuration – e. g. 1
Decisions:	e. g. submit for the first activity or “admin” and “development” for the second activity
Description:	some text to understand the reason for this activity
Inbox notifcation:	Additional we can enable the “inbox notification” for this activity – but this makes no sense for the first activity
Owners / Readers:	Finally we can select the activity owners and readers from the configuration with the actions “Owner” and “Reader”

For our example we use the following settings:

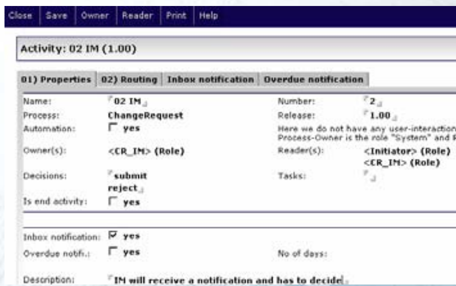
Name	Decisions	Owner	Reader	Inbox
01 Creation	submit	<Initiator>	<Initiator>	NO
02 IM	submit Admin / Dev / reject	<CR_IM>	<CR_IM> <Initiator>	Yes
03 Admin	finish	<CR_Admin>	<CR_IM>, <Initiator>…	Yes
03 Dev	finish	<CR_Dev>	<CR_IM>, <Initiator>…	Yes
04 TW	Forward to end	<CR_TW>	All above	Yes
END	Is End Activity = yes	<System>	All above	Yes



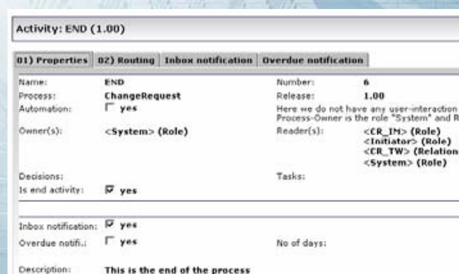
Process definition of our new process



First activity « 01 Creation »



Activity « 02 IM »



And finally the « END » activity for informing the user that this process has been finished

3.3.3 Routing

Now we can define the routing for each activity.
Therefore we have to open each activty and switch to the tab "02) Routing".
Based on the example of the activity "02 IM" we have 3 options:

a) The IM rejects the requests and sends it back to the user…
b) The IM forwards the request to the Admin department
c) The IM Forwards the request to the Dev department (group)

The following fields have to be entered for a proper routing definition:

Name:	will be displayed in the action
Next activity:	will define where to route to.. select from list of activities
Condition:	either select the name of the decision entered in the first tab or "formula" if you want to evaluate this condition only if a formula is true (like budget less than 1.000 EURO
Description:	will explain your user what this decision will do

3.3.4 Mail notifications

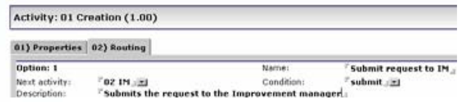
Within the system we can define mail notifications to inform participants of the workflow about new requests.
In the notification tabs of each activity we can enter the following fields:

Receiver:	select between all activity owners and readers
Principal:	enter a name here if you want to have a special sender like "CR-System
Subject:	Here you can enter a formula to evaluate the subject - @Formulas might be used
Body:	Here you can enter a formula to evaluate the body - @Formulas might be used

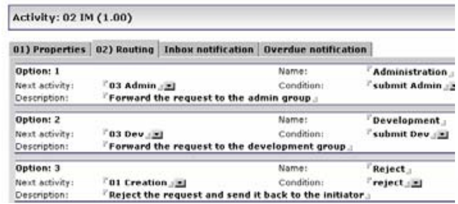
Here we can also reference to fields contained in our document like shown in the example below (fd_ShortDescription):

```
Subject:
"ChangeRequest: new request for approval " + fd_JobNameVFL

Body:
"Dear IM," + @NewLine +
"please approve my request about " + fd_ShortDescription + @NewLine +
@NewLine +
"Yours" + @NewLine +
@Name([CN];@UserName)
```

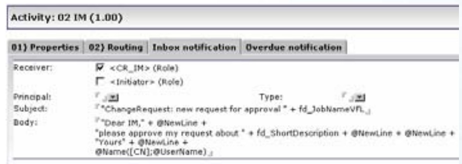


Routing option for « 01 Creation »



Routing options for the activity « 02 IM »

Routing options for « 03 Admin » and « 03 Dev »



Mail notification for the activity « 02 IM »

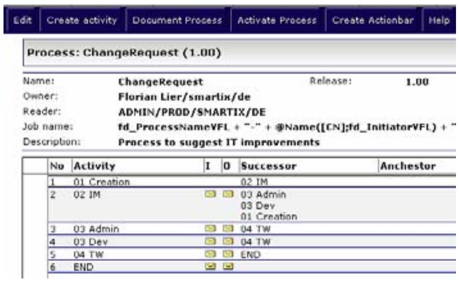
GMI KG · Gesellschaft für Migration und Integration
Bismarckstraße 81 · 70197 Stuttgart · Fon: +49 (0) 711 – 45 999 131
Fax: +49 (0) 711 – 45 999 135 · www.gesmi.de · info@gesmi.de

4 Deploy the workflow

4.1 Deploy the code / configuration

After finishing all steps described above we can deploy the workflow to our application. Here we have to click on 2 actions in the process document:

a) Activate Process (and select the target)
b) Create Actionbar (and select the target)

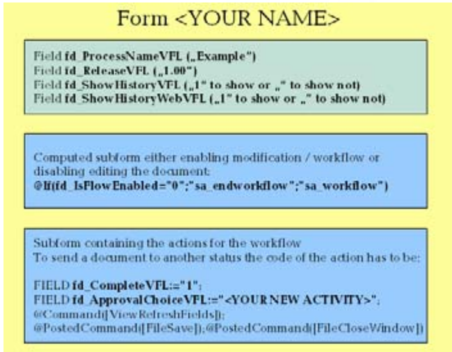


Actions for deploying the workflow

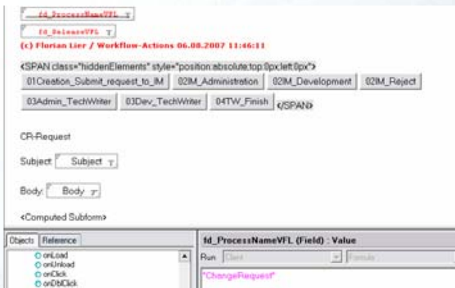
4.2 Include the code in the application

Finally we have to include the following elements in our form in the target database:

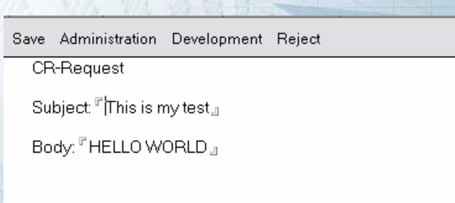
Fd_ProcessNameVFL:	"ChangeRequest" Add a computed field to your form containing the process name to the very top of the form
Fd_ReleaseVFL	"1.00" Add a computed field to the very top of your form containing the release number of the process
Fd_ShowHistoryVFL:	"1" Add a computed for display field to the top of your form if you want to show the workflow history
Fd_ShowHistoryWebVFL:	"1" Add a computed for display field to the top of your form if you want to show the workflow history in the web
ChangeRequest.Workflow Actions | sa_wfactions.ChangeRequest.1.00	Add the created subform containing the workflow actions to your form
Computed Subform	@If(fd_IsFlowEnabled="0";"sa_endworkflow";"sa_workflow") Add a computed subform to your form with the code above – this will include the subform containing the workflow code as long as your request is active… and the history otherwise



Elements to include in our form



Final form « ChangeRequest »



Form in activity / status « 02 IM »

System requirements :
Formula(r).WorkFlow runs with Lotus Notes Release 5, 6, 7 and 8.