



# Arco

Doma 6

Application administrator manual

11. Events and actions

**A new era in document management.**



---

# Doma 6

---

## Application administrator manual *11. Events and actions*

---

Document reference: not yet assigned  
*Described Doma version: 6.5*

---

Author: Arco Training Dept. - Kris Steenackers

---

### Manual Modification Follow-Up

Ref.	Date	Author	Modification description	Affected paragraphs
	26/07/2011	Kris Steenackers	Manual creation	

## Contents

1	Introduction .....	7
2	Configuration in the Doma admin module .....	8
2.1	Event actions .....	8
2.1.1	Definition .....	8
2.1.2	Events - overview .....	8
2.1.3	Actions – overview .....	9
2.1.4	Configuration of Event actions .....	10
2.2	User events .....	11
2.2.1	Definition .....	11
2.2.2	Events .....	11
2.2.3	Actions .....	11
2.2.4	Configuration of User events .....	11
2.3	Configuration of a condition for event actions and user events .....	13
2.3.1	Create a condition .....	13
2.3.2	Edit a condition .....	15
2.3.3	Delete a condition .....	15
3	Programming – configuration in the Doma admin module .....	16
3.1	Category type handler .....	16
3.2	Category event action: execute code .....	17
3.3	Screen handler .....	17
3.4	Batch Jobs .....	18
3.4.1	Definition .....	18
4	Overview and description of all actions .....	19
4.1	Category General .....	19
4.1.1	Cancel .....	19
4.1.2	Create object .....	20
4.1.3	Dispatch .....	21
4.1.4	Execute code .....	22
4.1.5	Execute User Event .....	23
4.1.6	Log .....	24
4.1.7	Move to folder .....	25
4.1.8	Perform Edit checks .....	26
4.1.9	Rename .....	27
4.1.10	Run in elevated mode .....	28
4.2	Category Notifications .....	29
4.2.1	Send notification to an assignee .....	29
4.2.2	Send notification to a mailinglist .....	30
4.2.3	Send notification to the case creator .....	31
4.2.4	Send notification to the work list .....	32
4.3	Category Properties .....	33
4.3.1	Add to property .....	33
4.3.2	Hide property .....	34
4.3.3	Remove from property .....	35
4.3.4	Set property .....	36
4.3.5	Set Runtime property .....	37
4.3.6	Show property .....	38
4.4	Category Template .....	39
4.4.1	Generate Template from Object .....	39
4.4.2	Generate Template .....	40
4.4.3	Generate Template .....	41
4.5	Category User Interface .....	42
4.5.1	Message box .....	42
4.5.2	Open File .....	43
4.5.3	Open Url .....	44
4.5.4	Refresh window .....	45
4.5.5	Send Mail .....	46

4.6	Category Versioning.....	47
4.6.1	Set new major version .....	47
4.6.2	Set new minor version .....	48
4.6.3	Set version .....	49
4.7	Category WorkFlow.....	50
4.7.1	Start Checkout Procedure.....	50
4.7.2	Start Procedure .....	51
4.7.3	Start Update Procedure .....	52
Appendix	Doma constants .....	53

DRAFT

## 1 Introduction

Though DocRoom is a standard product, customizing is possible on several levels.

**Application administrators** have the possibility to **customize** the DocRoom and Routing environment with the **configuration** of:

- **Event actions:** define an extra action that will be executed at a pre-defined moment in time.
- **User events** in the standard web interface. User events are pre-defined actions that are triggered by the user himself.

If you are familiar with **programming**, you can also create **handlers**.

In a handler, you define code; this code is programmed in dotnet and will be made available as an extra .dll file that has to be registered in the Doma admin module.

Next to these handlers, it is also possible to create your own **batch jobs** that run in the background of the application.

This document describes the configuration of these customization possibilities.

If you would like to know more about programming for DocRoom and Routing, please read the **Doma 6 Programming Guide**.

All manuals and guides can be found on [www.docroom.net](http://www.docroom.net).

## 2 Configuration in the Doma admin module

### 2.1 Event actions

#### 2.1.1 Definition

Event actions are pre-defined actions that can be activated at one specific moment (event). They can be configured in the Doma admin module on object or procedure level by a Doma administrator. Since these actions are pre-defined, the Doma administrator only has to configure an event action instead of having to program it.

#### 2.1.2 Events - overview

An *event* is the moment when an action is executed. They can be selected from a pre-defined list during configuration.

##### 2.1.2.1 Overview of the events for Folder, Document and Case categories

These are the events that are available for Folder, Document and Case categories:

Event	Description	Supports cancel?
<b>OnBeforeInsert</b>	Raised before an object reference is created in the database.	<b>Yes</b>
<b>OnAfterInsert</b>	Raised after an object reference is created in the database. This is still before the user sees the insert screen.	<b>No</b>
<b>OnBeforeSave</b>	Raised before the actual save of an object to the database. Use this event to implement specific business rules for the category.	<b>Yes</b>
<b>OnAfterSave</b>	Raised after the succesfull save of an object to the database. Use this event to notify external system of the succesfull save of an object to DocRoom.	<b>No</b>
<b>onComplete-Creation</b>	Raised after the successfull creation of the object by the end user (once all mandatory fields are entered).	<b>Yes</b>
<b>OnAfterDrum-Insert</b>	When an object is added via DRUM, this event allows you to add specific logic to this event. Worden de andere events nog getriggered??	<b>No</b>
<b>OnBeforeFileAdd</b>	Whenever a file is added to an object this event is raised to allow developers to implement custom checks on this file. The file addition can be cancelled by developers.	<b>Yes</b>
<b>OnFileIndex-Complete</b>	After completion of the default indexing of a file, this event is raised to implement further indexing or postprocessing options.	<b>No</b>
<b>OnLock</b>	Raised when a user wants to lock an object. The lock can be cancelled through this event.	<b>Yes</b>
<b>OnUnlock</b>	Raised when a user unlocks an object. This event cannot be cancelled.	<b>Yes</b>
<b>OnCheckOut</b>		<b>Yes</b>
<b>OnCheckIn</b>		<b>Yes</b>



<b>OnBeforeMove</b>	Before a user moves a document, extra checks can be implemented in this event.	<b>Yes</b>
<b>OnAfterMove</b>	After a succesfull move, post processing options can be implemented in this event (f.i. notify the owner of a dossier that someone moved a document in his folder)	<b>No</b>
<b>OnExpiry</b>	This event is raised when the expiry date of an object is passed.	<b>No</b>
<b>OnDelete</b>	Before a document is deleted, this event is raised. The deletion can be cancelled in this event.	<b>Yes</b>

### 2.1.2.2 Overview of the events for Procedures and Case categories

These are the events that are available for procedures and Case categories:


Event	Description
<b>WF_Entry</b>	Whenever a case enters a step, this event is raised.
<b>WF_Keep</b>	Every time a user saves a case, this event is raised. <i>Note:</i> When you activate the parameter 'Refresh on change' for a property and the property is changed, this event is also called.
<b>WF_Step_Expiry</b>	When a step expires (due date is reached), this event is raised.
<b>WF_Exit</b>	When a user presses the release button, this event is raised. <i>Note:</i> it is possible to prevent the dispatch by filling out the reject comment of the case.
<b>WF_Case_Expiry</b>	When the entire procedure reaches its due date, this event is raised.

### 2.1.3 Actions – overview

Event actions and user events can trigger pre-defined actions.

See also chapter **Overview and description of all actions**.

### 2.1.4 Configuration of Event actions

1. Start the Doma admin module.
2. Open the category where you want to create an event action.
3. Select the tab Event Actions.
4. Click on the button **New Action**  in the toolbar.
5. A pop-up window appears. It contains an overview of all actions that can be created, organized per category.
6. Select the action you want to create, e.g. **Set property** (category: Properties).
7. The action configuration screen appears.


EVENT	ACTION	NOT	CONDITION
WF_Entry ▾	<b>Set Property Requested by...</b>	<input type="checkbox"/>	Always
Property ▾	Requested by		
Value	#USER_STARTER#		

- a. **Event:** select the event when the action has to be activated in the dropdown list (click in the field to show the list).
- b. **Action:** selected action (in this example **Set property**) + the property name on which the action is active.
- c. **Not:** checkbox linked to the condition at the right.
- d. **Condition:** By default Always, but you can add one or more conditions that define when this action must or must not be executed.

CONDITION
Always ▾ + × ...

- e. **Property:** Property on which the action is executed; click in the field to select one from the list. The list shows all properties defined for the selected object/procedure.
- f. **Value:** value that has to be set; this value can be
  - Text
  - Reference to a property value [@property name@]
  - Reference top fixed properties, e.g. #USER\_STARTER#



8. Click on the Save button  in the toolbar to save the configuration.

## 2.2 User events

### 2.2.1 Definition

*User events* are pre-defined custom actions that users can execute in the web interface (either as a toolbar button in detail screens or as a right mouse click in result lists).

They can be configured in the Doma admin module on object or procedure level by a Doma administrator. Since these actions are pre-defined, the Doma administrator only has to configure an event action instead of having to program it.

The user events are implemented within the security context of DocRoom and Routing and can therefore be individually secured in the standard ACL management interface (a separate securable action is available for every defined user event).

### 2.2.2 Events

The event of a user action is when a user clicks on a toolbar button or on an item in the context menu. Both button and item can to be configured in the Doma admin module on category or procedure configuration level.

### 2.2.3 Actions

Event actions and user events can trigger pre-defined actions.  
*See also chapter **Overview and description of all actions**.*


### 2.2.4 Configuration of User events

#### 2.2.4.1 In general


The configuration of a user event is done in 2 steps:


1. Creation of a button or context menu item (= event)
2. Link one or more actions to the event.

#### 2.2.4.2 Detailed example

1. Start the Doma admin module.
2. Open the category where you want to create a user event.
3. Select the tab **User Events**.
4. Click on the button **New Event**  in the toolbar.
5. A new link appears in the overview. Define the event here.

	NAME	ICON	LOCATION
>	Enter text here		ScreenToolbar, ContextMenu

- a. **Name:** Name of the user event. This name appears in the contextmenu and in the toolbar as a toolbar button.
- b. **Icon:** Icon that is shown in the toolbar, click on the button  to show the folder where the icons are located. Then select an icon.
- c. **Location:** place where the user event *can be shown*: in the toolbar of the screen or in the context menu of the object or both. (To really show this user event, it has to be selected in the screen configuration.)

6. Now click on the button **New Action**  in the toolbar.
7. A pop-up window appears. It contains an overview of all actions that can be created, organized per category.
8. Select the action you want to create, e.g. **Send notification to the case creator** (category: Notifications).
9. The action configuration screen appears:

NAME	ICON	LOCATION
> Send feedback		ScreenToolbar, ContextMenu


  

ACTION	NOT	CONDITION
> Send notification to the...	<input type="checkbox"/>	Always
> NotificationTemplate		Feedback

- a. **Action:** selected action (in this example **Set property**) + the property name on which the action is active.
- b. **Not:** checkbox linked to the condition at the right.
- c. **Condition:** By default Always, but you can add one or more conditions that define when this action must or must not be executed.

CONDITION
Always ▼ + X ...

- d. **Notification template:** Notification template used to send the information. Click in the field to select one from the list. The list shows all notification templates defined on application level.

10. Click on the Save button  in the toolbar to save the configuration.

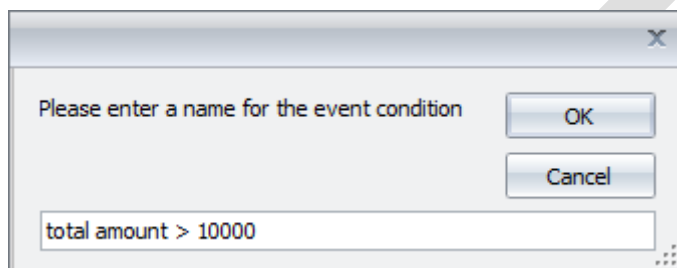
## 2.3 Configuration of a condition for event actions and user events

### 2.3.1 Create a condition

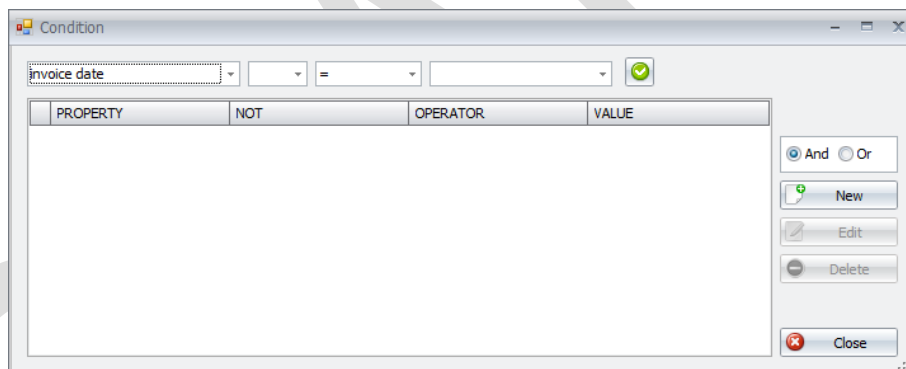
1. Click on the Add icon  in the condition overview to add a new condition:



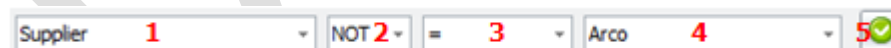
2. A new window appears – enter the name of the condition and click on the OK button:



3. The condition configuration window appears.



4. Define your condition(s):



- a. Select the property on which the condition is based (1).
- b. Define the condition in the fields 2 (not), 3 (operator) and 4 (value).
- c. Add the condition to the overview by clicking on 5.

Some examples:

Supplier	NOT	=	Arco	✓
Total amount		>	10000	✓
Total amount		BETWEEN	5000 and 10000	✓

5. The condition appears in the overview:

	New	Click here to add a new condition to the overview.
	Edit	Click here to edit the condition selected in the overview.
	Delete	Click here to delete the condition selected in the overview.
	Close	Click here to close the overview.

When more than one condition has been created, they are combined with the AND or OR option:

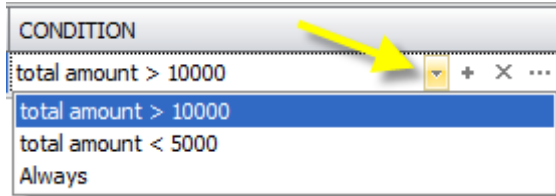
AND: both conditions have to be met.  
OR: only one of the conditions has to be met.


6. The newly created condition now appears in the overview:

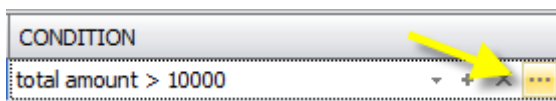
CONDITION
total amount > 10000

### 2.3.2 Edit a condition

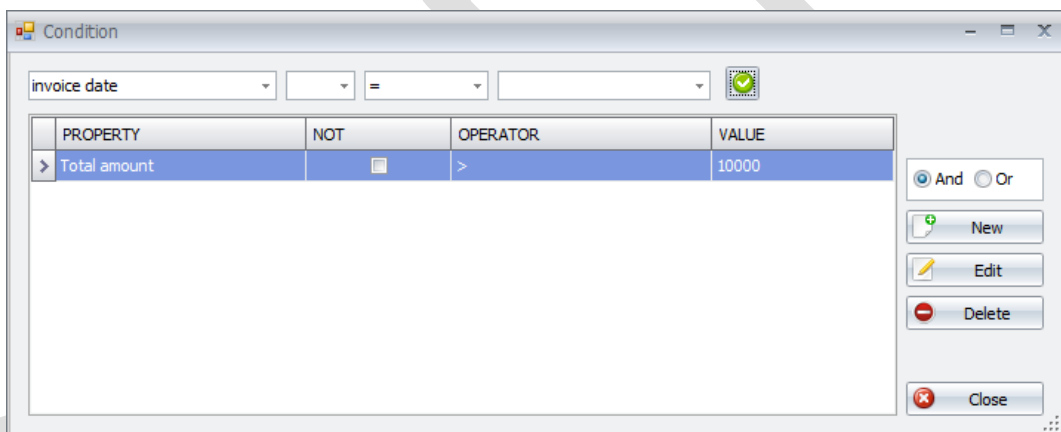
1. Click on the show list icon  to show all conditions:



2. Select the condition you want to edit and click on the edit icon  :



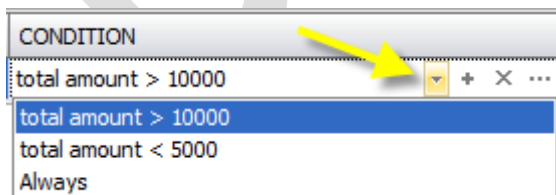
3. The condition configuration window opens:



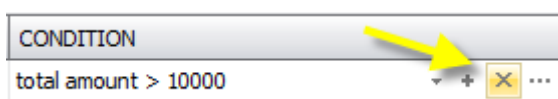
4. Edit your condition.

### 2.3.3 Delete a condition

1. Click on the show list icon  to show all conditions:



2. Select the condition you want to delete and click on the delete icon  :



### 3 Programming – configuration in the Doma admin module

#### 3.1 Category type handler

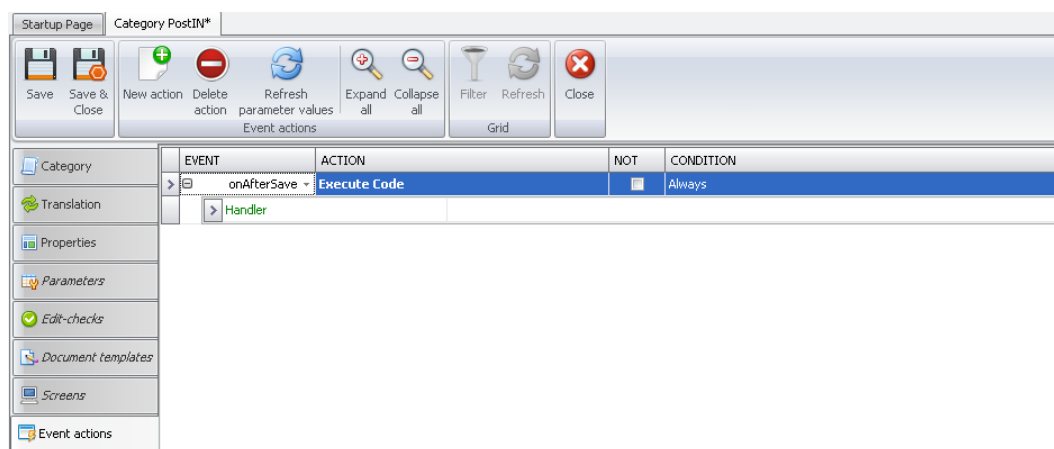
Category type handlers are handlers that can be linked to one or more Doma categories. The category type handler can be linked in the Category configuration tab in the Doma admin module: the field handler will show a dropdown list with all handlers for the selected Doma category.

The screenshot displays the 'Category' configuration window in the Doma admin module. The window has a top toolbar with icons for Save, Save & Close, New, Delete, Export, and Close. The left sidebar lists various configuration categories. The main panel is divided into two sections: 'Category' and 'Category type handler'. The 'Category' section includes fields for ID, Name, Icon, Type, Create documents in folder, Auto name, File cardinality, and a checkbox for 'Disable manual creation'. The 'Category type handler' section features a dropdown menu for 'Handler'.

Administration of all category type handlers is done in the item Category type handlers of the part Settings of the Doma admin module.



### 3.2 Category event action: execute code



### 3.3 Screen handler

The (detail) screens that are used in the WebInterface can be configured in the Doma Admin module. The items on the screen, their position and mode as well as the toolbar of the screen can be configured. In certain cases you will want to extend this default:

- the mode of a field is conditional (if in a list the option other is selected, then the reason field is mandatory, in other cases this is optional)
- a number of fields only have to be visible if a specific selection is made or a specific type of users is looking at the data, in all other cases they should not be shown to the user.
- You want to add HTML information blocks that show real time information of external data to the user without synchronizing the data to the DocRoom database.
- The toolbar buttons should only be shown when certain conditions are met (even if the user has enough rights to see them)
- ...

**Note:** screen handlers are executed on the webserver just before rendering the screen to the user. When a user enters data and sends it back to the server, DocRoom has to execute the screenhandler again in order for DocRoom to be able to recover the data the user entered on this screen.

## 3.4 Batch Jobs

### 3.4.1 Definition

Batch jobs are asynchronous jobs that are executed in the batch queuing context of DocRoom and can be scheduled/executed/monitored in the same way as the default DocRoom and Routing Batch jobs.

*Please see the Doma 6 programming guide for detailed information about this subject.*

The screenshot displays the 'Screen Result screen invoices' configuration window. The toolbar at the top includes icons for 'Save', 'Save & Close', 'New', 'Delete', 'Security', 'Export', and 'Close'. The main configuration area contains the following fields:

- Name:** Result screen invoices
- ID:** 110960
- Category:** Invoice
- Procedure:** (empty field)
- Screen mode:** RESULT (dropdown menu)
- Screen type:** DefaultMode (dropdown menu)
- Screen source:** CustomResultGrid\_110960.xml
- Custom handler:**
  - Assembly name:** (empty field with expand/collapse icons)
  - Class name:** (empty field with expand/collapse icons)

## 4 Overview and description of all actions

### 4.1 Category General

#### 4.1.1 Cancel

Category General

Can be used at several events to cancel

- the creation of a document, folder, case.
- the dispatch on a normal step.

Can be use with all **on before** events (objects) and **on exit** events (procedure).

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

	ACTION	NOT	CONDITION
>	Cancel action canceled	<input type="checkbox"/>	Always
	> Message		

**Message** Message that will appear after cancel.

### 4.1.2 Create object

Category General

Will create a new DocRoom object.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

Configuration screen for objects:

ACTION	NOT	CONDITION
> Create Object All properties	<input type="checkbox"/>	Always
Category		All properties
Object Name		
> New window		
Refresh		

Configuration screen for procedures:

ACTION	NOT	CONDITION
> Create Object	<input type="checkbox"/>	Always
> Category		
Object Name		
Package		
New window		
Refresh		

**Category** List of all existing Doma Categories. Click in the field to show the list. Select one of the items in the list. The newly created object will be one of the selected category.

**Object Name** Name of the created object.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

**Package** Package where the object has to be linked. (Only when creating from a procedure.)

**New window** Open the new object in a new window yes or no. Click in the field to select a value.

**Yes** New object creation window will open in a new window.

**No** New object creation window will open in the current window, so detail window of the object where you were is replaced by the configuration window of the new object.

**Refresh** Refresh the current window yes or no. Click in the field to select a value.

**Yes** Current window will be refreshed after close of the message box.

**No** Current window will not be refreshed.

### 4.1.3 Dispatch

Category General

Will dispatch the current step of the procedure.

Available as an event action on Case Category and Procedure level. Available as a user event.

	ACTION	NOT	CONDITION
> ⊕	Dispatch	<input type="checkbox"/>	Always

#### 4.1.4 Execute code

Category General

Will execute code that is specifically written for this action.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

Configuration screen for objects:

ACTION	NOT	CONDITION
> Execute Code	<input type="checkbox"/>	Always
> Handler		

Configuration screen for procedures:

ACTION	NOT	CONDITION
> Execute code	<input type="checkbox"/>	Always
> Script		

**Handler** Select the handler that has to be executed by clicking in the field.  
Administration of the available handlers is done in the item *Category event handlers* in the settings part of the Doma admin module.

**Script** Select the script that has to be executed by clicking in the field.  
Administration of the available handlers is done in the item *Routing event handlers* in the settings part of the Doma admin module.

#### 4.1.5 Execute User Event

Category General

Will automatically execute the action(s) linked to the selected user event.

*In a User event, it is possible to add more than one action. All those actions are triggered by this one event and will be executed in one block. This option allows you to do the same automatically.*

*Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.*

ACTION	NOT	CONDITION
> <b>Execute User Event</b>	<input type="checkbox"/>	Always
<input type="text" value="User Event"/>		<input type="text" value="Send feedback"/>

**User Event** Select the User Event that has to be executed by clicking in the field.

#### 4.1.6 Log

Category General

Logs text to the global logfile.

Default location of the file: D:\Arco\Logging.

Filename: GlobalLog[YYYYMMDD].txt

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
>  Log	<input type="checkbox"/>	Always
>  Text		

##### Text

Text that will be logged in the log file.

This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.



#### 4.1.7 Move to folder

Category General

The object will be moved to another folder. If the folder where the document has to be moved to doesn't exist, it will be created.

*If the object is moved to a new folder that cannot be created, the document will not be moved.*

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <b>Move to folder</b>	<input type="checkbox"/>	Always
> Folder		
Category for new folders		

##### Folder

Name of the folder where the item has to be moved to.  
Insert the complete path to the folder.  
e.g. invoice\2011\07 to ensure that the item is moved to the folder 07, which is a subfolder of the folder 2011, which is a subfolder of the folder invoice, which is a subfolder of the root folder.

This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see appendix)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add  
=[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.

An inline compiler: start with ?, then add your Csharp code.

##### Category for new folders

Category of the new folder that is created when the mentioned folder does not exist.

#### 4.1.8 Perform Edit checks

Category General

Will perform the selected edit checks. When an edit check returns one or more errors, the item will not be saved.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> <b>Perform Edit Checks</b>	<input type="checkbox"/>	Always
<div> <div>&gt;</div> <div>Query Type</div> </div>		
<div> <div></div> <div>Group</div> </div>		
<div> <div></div> <div>Level</div> </div>		

**Query Type**

**Group**

**Level**

#### 4.1.9 Rename

Category General

Renames the object/case.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> <b>Rename</b>	<input type="checkbox"/>	Always
<input type="checkbox"/> > Name		

##### Name

New name for the object.

This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1@]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

#### 4.1.10 Run in elevated mode

Category General

Switches the context to elevated mode: the action will overrule the default security rights and code will be executed with full control rights. After execution of the event, elevated mode will be ended, so default security will be active again.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

	ACTION	NOT	CONDITION
> +	Run in elevated mode	<input checked="" type="checkbox"/>	Always

## 4.2 Category Notifications

### 4.2.1 Send notification to an assignee

Category General

Send an automatic e-mail to the user(s) selected in an assignee property or to the e-mail address of an e-mail property.

Available as an event action on Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> <b>Send a notification to an assignee</b>	<input type="checkbox"/>	Always
<input type="checkbox"/> NotificationTemplate		
> Assignee		

#### NotificationTemplate

Notification template that has to be used. Click in the field to see the list of notification templates.

*Notification templates can be created in the item Configuration Templates in the Configuration part of the Doma admin module.*

#### Assignee

Assignee to whom the notification has to be sent. Click in the field to see the list of available properties of the type assignee and/or e-mail.

*Properties of the type Assignee or E-mail can be created as object property or as pool property in the Configuration part of the Doma admin module.*

## 4.2.2 Send notification to a mailinglist

Category General

*Not yet implemented.*

Available as an event action on Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <b>Send notification to a mailinglist</b>	<input type="checkbox"/>	Always
> NotificationTemplate		
> MailingList		

### Notification Template

Notification template that has to be used. Click in the field to see the list of notification templates.

*Notification templates can be created in the item Configuration Templates in the Configuration part of the Doma admin module.*

### Mailinglist

### 4.2.3 Send notification to the case creator

Category General

Send an automatic e-mail to the creator of the dossier.

Available as an event action on Case Category and Procedure level. Available as a user event.

	ACTION	NOT	CONDITION
>	Send notification to the case creator	<input type="checkbox"/>	Always
	> NotificationTemplate		

#### Notification Template

Notification template that has to be used. Click in the field to see the list of notification templates.

*Notification templates can be created in the item Configuration Templates in the Configuration part of the Doma admin module.*

#### 4.2.4 Send notification to the work list

Category General

Send an automatic e-mail to the step executors.

Available as an event action on Case Category and Procedure level. Available as a user event.

	ACTION	NOT	CONDITION
>	Send notification to the worklist	<input type="checkbox"/>	Always
	NotificationTemplate		

##### Notification Template

Notification template that has to be used. Click in the field to see the list of notification templates.  
*Notification templates can be created in the item Configuration Templates in the Configuration part of the Doma admin module.*



## 4.3 Category Properties

### 4.3.1 Add to property

Category Properties

To be used in a multi select property.

This action will check whether the new value is already present in the multiselect property.

If the value does not exist, it will be added to the list.

If the value does exist, the value is not added.

If this action is used on a property without multi select, it will act like the *Set property* action.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
Perform Edit Checks	<input type="checkbox"/>	Always
Add to property	<input checked="" type="checkbox"/>	Always
<div> <div>&gt;</div> <div>Property</div> </div> <div> <div>&gt;</div> <div>Value</div> </div>		

**Property** Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.

**Value** Value that will be added to the selected property.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1@]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

### 4.3.2 Hide property

Category Properties

Don't show the property in the web interface screen.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> <b>Hide Property</b>	<input type="checkbox"/>	Always
<input type="checkbox"/> > Property		

**Property** Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.

### 4.3.3 Remove from property

Category Properties

To be used in a multi select property.

This action will check whether the value is present in the multiselect property and will remove it from the list if the value exists.

If this action is used on a property without multi select, it will act like the *Set property* action.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> Remove from property	<input type="checkbox"/>	Always
> Property		
Value		

**Property** Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.

**Value** Value that has to be removed from the property when the action is executed.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1@]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

#### 4.3.4 Set property

Category Properties

Will put a value in the selected property.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <b>Set Property Reason</b>	<input type="checkbox"/>	Always
Property		Reason
> Value		demo

**Property** Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.

**Value** Value that appears in the property when the action is executed.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1@]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

### 4.3.5 Set Runtime property

Category Properties

By setting runtime properties, you can declare local variables. These can be used to make calculations. Only the result of your calculation will be written to the database. The local variables are deleted after processing.

E.g. on load (of an object) alle runtime props are initialized, so this information doesn't have to be written to the database. Runtime properties are then used to make some calculations. Results are copied to one or more properties of the object. The local variables are deleted after processing.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <b>Set Runtime Property</b>	<input type="checkbox"/>	Always
<div> <div>&gt; Property</div> <div>Value</div> </div>		

**Property** Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.

**Value** Value of the runtime property.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1@]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

### 4.3.6 Show property

Category Properties

Show the property in the web interface screen.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <input type="checkbox"/> <b>Show Property</b>	<input type="checkbox"/>	Always
<input type="checkbox"/> > Property		

#### Property

Select a property from the list that appears when you click on the field. The list shows all properties of the object/procedure.  
*You can select more than one property in this config.*

## 4.4 Category Template

### 4.4.1 Generate Template from Object



*Category Template*

Will automatically generate a file from a template.

- When generated on **document level**, only a file can be created from the template. This file will be added to the document where it was created.

*Available as an event action on Document Category.*

Configuration screen for objects:

ACTION	NOT	CONDITION
>  <b>Generate Template from Object</b> ALLPROPS	<input type="checkbox"/>	Always
>  Template		ALLPROPS

**Template**      Template that is to be used to create the new file.

#### 4.4.2 Generate Template

Category Template

Will automatically generate a document or a file from template. The created item is then linked to the selected package.

Available as an event action on Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
> <b>Generate Template</b>	<input type="checkbox"/>	Always
> Template		
Create		
Package		

**Template** Template that is to be used to create the new file.

**Create** Document or file. (Only available with procedures.) Define whether only a file or a document with the file as an attachment is created.  
*The document category that will be used is defined on document template level. Configuration of document templates is done in the item Document templates of the configuration part of the Doma admin module.*

**Package** Package where the new document or file will be pasted. (Only available with procedures.)  
*When using a case category, the file will be linked directly to the object. When using an insert procedure, then file will be linked to the target object.*



#### 4.4.3 Generate Template

Category Template

Generates a File or object from a case **with User Interface** (UI).

*Since the user interface is used, this action can only be started from the user interface.*

Available as a user event.

ACTION	NOT	CONDITION
> <b>Generate Template</b>	<input type="checkbox"/>	Always
> Template		
Open in		

**Template** Template that is to be used to create the new file.

**Open in** Frame/window where the URL will be opened:

- **Hidden Frame** Open the template is a hidden window. (This has no advantages for this action.)
- **New Window** URL is opened in a new window.
- **Preview Window** URL is opened in the preview window at the right of the screen.
- **Same Window** URL is opened in the same window

## 4.5 Category User Interface

### 4.5.1 Message box

Category User Interface

Show a message box.

Available as a user event.

	ACTION	NOT	CONDITION
>	<b>MessageBox</b>	<input type="checkbox"/>	Always
	> Message		
	Refresh		

**Message** Message that appears in the message box.

**Refresh** Refresh the current window yes or no. Click in the field to select a value.

- Yes** Current window will be refreshed after close of the message box.
- No** Current window will not be refreshed.

## 4.5.2 Open File

Category User Interface

Open a file on the server.

Available as a user event.

ACTION	NOT	CONDITION
> <b>Open File</b> D:\Arco\Data\Ro...	<input type="checkbox"/>	Always
> File		D:\Arco\Data\RoutingDocs\rpwd.txt
Download		False
Refresh		True

**File** Name of the file. (+ full path on the server)

**Download** Download the file or open it in DocRoom's default file viewer. Click in the field to select a value.

**True** File will be downloaded to the PC of the user.

**False** File is opened with the default DocRoom viewer.

**Refresh** Refresh the details screen from where you opened the file. Click in the field to select a value.

**True** Detail screen will be refreshed from the moment the file is shown. (use this when the action changes items in the detail screen)

**False** Detail screen is not refreshed.

### 4.5.3 Open Url

Category User Interface

Open a URL.

Available as a user event.

ACTION	NOT	CONDITION
> <b>Open Url</b>	<input type="checkbox"/>	Always
> Url		
Open in		
Refresh		

**Url** URL that is to be opened.

**Open in** Frame/window where the URL will be opened:

- **Hidden Frame** URL is opened in a hidden frame. User will not see it.
- **New Window** URL is opened in a new window.
- **Preview Window** URL is opened in the preview pane at the right part of the window.
- **Same Window** URL is opened in the same window

**Refresh** Refresh the details screen from where you opened the URL. Click in the field to select a value.

**True** Detail screen will be refreshed from the moment the URL is shown.

**False** Detail screen is not refreshed.

#### 4.5.4 Refresh window

Category User Interface

Refreshes the current window.  
Use this action the show new values in the screen.

Available as a user event.

	ACTION	NOT	CONDITION
> +	Refresh Window	<input type="checkbox"/>	Always

### 4.5.5 Send Mail

Category User Interface

Opens the mail window to send a mail message.

Available as a user event.

ACTION	NOT	CONDITION
> <b>Send Mail</b>	<input type="checkbox"/>	Always
> Template		
Link as		
Refresh		

**Template** Template that is to be used for the mail message. Click in the field to show the existing templates. The templates are defined in the Doma admin module. (notification templates)

**Link as** The mail can be linked to the object from where it is started in different ways:

<b>Comment</b>	As a mail comment to the object.
<b>Don't link</b>	Not linked.
<b>File</b>	As a file to the object.

**Refresh** Refresh the details screen. Click in the field to select a value.

**True** Detail screen will be refreshed from the moment the URL is shown.  
**False** Detail screen is not refreshed.

## 4.6 Category Versioning

### 4.6.1 Set new major version

Category Versioning

Changes the current document version to a new major version. (The first digit of the version number will be augmented.)

*Can be used at the end of a check-out procedure to change the version. (Version is not automatically changed when using a check-out procedure.)*

*Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.*

ACTION	NOT	CONDITION
> + Set new major version	<input type="checkbox"/>	Always


#### 4.6.2 Set new minor version

Category Versioning

Changes the current document version to a new minor version. (The second digit of the version number will be augmented.)

*Can be used at the end of a check-out procedure to change the version. (Version is not automatically changed when using a check-out procedure.)*

*Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.*

ACTION	NOT	CONDITION
>  <b>Set new minor version</b>	<input type="checkbox"/>	Always



### 4.6.3 Set version

Category Versioning

Changes the current document version to a new version. This new version is defined in the set version configuration.

*Can be used at the end of a check-out procedure to change the version. (Version is not automatically changed when using a check-out procedure.)*

*Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.*

WF_Case_Expiry ▾	<b>Set version</b>	<input type="checkbox"/>	Always
<div> <div>&gt;</div> <div>Version</div> </div>			

**Version** Version number that has to be entered. Must be of the format N.N.N, e.g. 1.2.0.

## 4.7 Category WorkFlow

### 4.7.1 Start Checkout Procedure

Category WorkFlow

Start a check-out procedure for the selected object.

Available as an event action on Document Category. Available as a user event.

ACTION	NOT	CONDITION
> <b>Start Checkout Procedure</b>	<input type="checkbox"/>	Always
<div> <div>Procedure</div> <div>&gt; Case Name</div> <div>Refresh</div> </div>		

**Procedure** List of available update procedures for the object.

**Case Name** Name of the case.  
This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see *appendix*)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

**Refresh** Refresh the details screen. Click in the field to select a value.

**True** Detail screen will be refreshed from the moment the URL is shown. *If refresh = true, the update procedure will open in the same window.*

**False** Detail screen is not refreshed.

## 4.7.2 Start Procedure

Category WorkFlow

Start a procedure from an object.

Available as an event action on Document Category, Folder Category, Case Category and Procedure level. Available as a user event.

ACTION	NOT	CONDITION
<input checked="" type="checkbox"/> <b>Start Procedure</b>	<input type="checkbox"/>	Always
<input checked="" type="checkbox"/> Procedure <input type="checkbox"/> Case Name <input type="checkbox"/> New window <input type="checkbox"/> Attach current object <input type="checkbox"/> Refresh		

### Procedure

List van update procs voor het item

### Case Name

Name of the case.

This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see appendix)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

### New window

Open the procedure in a new window? Click in the field to select a value.

#### True

Procedure detail will open in a new window.

#### False

Procedure detail will open in the same window. (So detail screen of the object from where you start the procedure is replaced by the detail screen of the procedure.)

### Attach current object

Will link the current object to the procedure. Click in the field to select a value.

#### True

Current object is linked to the procedure.

#### False

Current object is not linked to the procedure.

### Refresh

Refresh the details screen. Click in the field to select a value.

#### True

Detail screen will be refreshed from the moment the URL is shown.

#### False

Detail screen is not refreshed.

### 4.7.3 Start Update Procedure

Category Workflow

Start an update procedure for the selected object.

*When edit of an object is not allowed, you can still allow an update by starting a user event where this action is linked to.*

Available as an event action on Document Category and Folder Category.

EVENT	ACTION	NOT	CONDITION
onAfterSave	Start Checkout Procedure ...	<input type="checkbox"/>	Always
Procedure			
Case Name			
Refresh			

#### Procedure

List van update procs voor het item

#### Case Name

Name of the case.

This value can be

- Plain text
- A reference to a property value [@property name@]
- A reference to fixed properties, e.g. #USER\_STARTER# (see appendix)
- A formula: start the value with =, then add your mathematical formula, e.g. to add the sum of 2 properties, add =[@nameOfProp1 @]+[@nameOfProp2]. Replace will be done before formula.
- An inline compiler: start with ?, then add your Csharp code.

#### Refresh

Refresh the details screen. Click in the field to select a value.

- True** Detail screen will be refreshed from the moment the URL is shown. *If refresh = true, the update procedure will open in the same window.*
- False** Detail screen is not refreshed.

## Appendix Doma constants

Constant	Description
<b>General constants</b>	
#LANGCODE#	Language code of the logged in user
#USER_LOGIN#	Account of the logged in user
#USER_GROUPS#	List of all groups to which the logged in user belongs
#USER_ROLES#	List of all roles to which the logged in user belongs
<b>Routing workflow constants</b>	
#TECH_ID#	Instance ID of the workflow case
#CASE_ID#	ID of the workflow case (the same for all instances of a workflow case)
#CASE_NAME#	Name of the workflow case
#STEP_EXECUTOR#, #USER_STEPEXECUTOR#	Account of the person who opened (and locked) the case for editing. This is the person who is actually working on the workflow case.
#USER_STARTER#	Account of the person (or process) who started the workflow case.
#DOSSIER_ID#	ID of the dossier in which the workflow cases are located.
#CASE_DESCRIPTION#	Description of the workflow case
#CASE_LISTINFO#	Listinfo parameter for the workflow case
#CASE_LISTINFO2#	Listinfo2 parameter for the workflow case
#CASE_LISTINFO3#	Listinfo3 parameter for the workflow case
#STEP_NAME#	Name of the step in which the case instance is located.
#PROC_NAME#	Name of the procedure of the workflow case.
#STEP_DUE#	Due date of the current step for the current case
#CASE_DUE#	Overall due date of the current case
#STEP_START#	Start date of the current step for the current case
#CASE_START#	Start date of the current case
<b>Object constants</b>	
#OBJ_ID#	Object ID of the current object
#OBJ_DIN#	Object DIN of the current object
#PARENT_OBJ_ID#	ID of the folder in which the object is located
#CAT_ID#	ID of the category of the current object
#CASE_ID#	Current case_id
#OBJ_NAME#	Name of the current object
#PARENT_NAME#, #FOLDER_NAME#	Name of the folder in which the object is located
#PARENT_PATH#, #FOLDER_PATH#	Entire path of the folder in which the object sits
<b>Date reference</b>	
#DATE_NOW#	Current date and time (YYYY-MM-DD HH:MM:SS)