# Nova Bonita Workflow

## Console User Guide

Nova Bonita Workflow

# Nova Bonita Workflow

## Console User Guide

Nova Bonita (aka Bonita v4)

Software

September2008

# Table of Contents


**Table of Contents** ..... iii

**List of Figures** ..... v

**Chapter 1.Overview** ..... 7

1.1Role of User ..... 8

1.2Role of Operator ..... 8

**Chapter 2.Installation** ..... 9

2.1Prerequisite ..... 9

2.2Installation procedure ..... 9

**Chapter 3.Quick Start** ..... 10

3.1Console Start ..... 10

3.2Deploy process ..... 10

3.3Start Process ..... 11

**Chapter 4.Process Console Description** ..... 12

4.1Console Access ..... 12

4.2Default users ..... 12

4.3Console frames description ..... 13

4.4Application's graphical organization description ..... 14

**Chapter 5.Accessing and Creating Processes** ..... 16

5.1Access the Workflow Process List ..... 16

5.2Create a new Instance of a Bonita Process ..... 16

**Chapter 6.Access To Do / Done Tasks** ..... 17

6.1Consult the To Do Tasks List ..... 17

6.2Perform / Pause / Resume a Task ..... 19

6.2.1Perform ..... 19

6.2.2Suspend ..... 19

6.2.3Resume ..... 20

6.3Consult the Done Tasks List ..... 20

**Chapter 7.Managing Process Models** ..... 21

7.1Access the Process Model List ..... 21

7.2Deploy / Undeploy / Delete Processes Models ..... 23

7.3Start Process Models....23
7.4Remove all Instances of a Process Model....23
Chapter 8.Managing Instances....24
8.1Access the Process Instances List....24
8.2Consult / Edit the variables of an instance....25
8.3Access the activities list of an instance....26
Chapter 9.Managing Activities....27
9.1Access the Activities List....27
9.2Start an Activity....28
9.3Suspend an Activity....28
9.4Resume an Activity....29
9.5Access the Variables List of an Activity....29
Chapter 10.Managing users, groups and memberships....30
10.1Add a new user to the Nova Bonita Console....31
10.2Set roles/permission of a user to access to the console....31
Chapter 11.Forms customization....35
11.1Overview 35
11.2Forms.xml syntax....35
11.2.1Overview....35
11.2.2Tag list....35
11.2.2.1<forms/>....35
11.2.2.2<form/>....36
11.2.2.3<activity/>....36
11.2.2.4<resource-bundle/>....36
11.2.2.5<customized-view/>....37
11.2.2.6<variable/>....37
11.2.2.7<validator/>....39
11.2.2.8<property/>....39
11.2.3Data Validators list....40
11.2.3.1DateTime....40
11.2.3.2EmailAddress....41
11.2.3.3Expression....41
11.2.3.4Number....42
11.2.3.5NumberInRange....42
11.2.3.6PositiveNumber....43
11.2.3.7SpecialCharacter....43
11.2.3.8StringLength....43
11.2.3.9Float....44
11.3Internationalize your forms....45
11.3.1Overview....45
11.3.2Syntax....45
11.3.2.1Overview....45
11.3.2.2Key list....45

# List of Figures

figure 1: Login screen....10
figure 2: Process Deployed....10
figure 3: Choose a process to deploy....10
figure 4: Start Process Form....11
figure 5: The created instance....11
figure 6: Console login screen....12
figure 7: Console Description....13
figure 8: Application Description (1/3)....14
figure 9: Application Description (2/3)....14
figure 10: Application Description (3/3)....15
figure 11: Access Workflow Process List....16
figure 12: Creation of a new instance form example....16
figure 13: Classical view of the To Do List....17
figure 14: Advanced view for the To Do List....17
figure 15: Detailed view of Task....18
figure 16: Perform a task....19
figure 17: The Done Tasks List....20
figure 18: Process Model List....21
figure 19: Detailed view of a process model....22
figure 20: Consult the instances list for a given process....22
figure 21: Detailed View of an Instance....25
figure 22: Consult the variables of an instance....26
figure 23: Consult Activities of a given instance....26
figure 24: Detailed view of an activity....28
figure 25: Access the variables of an activity....29

figure 26: Access Administration page.....30

figure 27: Add new user to the Nova Bonita Console.....31

figure 28: Select the Bonita group.....32

figure 29: fill the Add member form to add as user.....33

figure 30: fill the Add member form to add as operator.....34

# Chapter 1. Overview

This document describes the process console capabilities for the Bonita Workflow. Although these activities may be performed by a single individual, the Process Console provides those capabilities to users in a role based basis. BPM User and Operator roles will be leveraged in this guide.

The information in this document is organized as follows:

### Process Console Description

Refer to Chapter 2.

### FOR THE USER

- Accessing and Creating Processes
  Refer to Chapter 3.
- Accessing and Managing To Do / Done Tasks
  Refer to Chapter 4.

### FOR THE OPERATOR

- Managing Process Models
  Refer to Chapter 5.
- Managing Instances
  Refer to Chapter 6.
- Managing Activities
  Refer to Chapter 7.

## 1.1 Role of User

This guide provides the User with the information necessary to be able to :

- Start Workflow Processes
- Perform / Suspend / Resume Tasks

## 1.2 Role of Operator

This guide provides the Operator with the information necessary to be able to :

- Deploy / Undeploy / Start Bonita process models
- Access Bonita process model instance informations
- Delete Instance
- Edit Process Instance Variables
- Access Bonita Activities informations
- Perform / Suspend / Resume Activity in a specific instance
- Consult / Edit Activity Variables
- Consult Activity's Properties

# Chapter 2. Installation

## 2.1 Prerequisite

The Nova Bonita Console works with java 1.5. Be sure that your default JRE is 1.5

## 2.2 Installation procedure

-Get the last version of the Nova Bonita Console from this web page :

http://forge.objectweb.org/project/showfiles.php?group_id=56

- Extract it and go in the **bonita-console-4.0/bin** directory

- If your use Linux as Operating system do the following::

> **chmod +x bonita-console-4.0/bin/*.sh**

> **unset CATALINA_HOME CATALINA_BASE**

# Chapter 3. Quick Start

In this chapter we present a quick start documentation for the Nova Bonita Console. In the next chapters we will explain in more details the functionalities available in this release

## 3.1 Console Start

- Open a command line and execute the following under the **bonita-console-2.0/bin** directory:

- For Linux : **./bpm.sh run**
- For Windows : **bpm run**

- In your web browser connect to the following URL : http://localhost:8080/portal/

Connect with :

- User Name : **root**
- Password : **bpm**

figure 1: Login screen

## 3.2 Deploy process

Open the **Bonita Management** application available in the doc bar.

A couple of BPM examples to deploy are available under **Bonita-console-4.0/examples/**.

Choose a process to deploy and click on

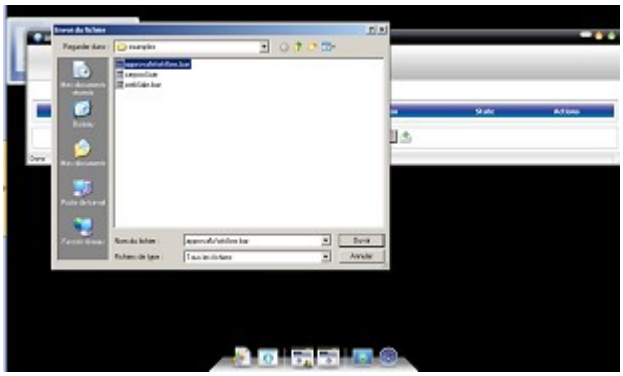


figure 3: Choose a process to deploy

figure 2: Process Deployed

## 3.3 Start Process

To start your first process click on the button

Fill the displayed form and click on the submit button.

So a new instance is created. And the updated instances list is displayed.

figure 4: Start Process Form

figure 5: The created instance

# Chapter 4. Process Console Description

## 4.1 Console Access

To access the Process Nova Bonita Console, connect to the following URL :

http://your_host:your_HttpPort/portal/

(by default : http://localhost:8080/portal/)

figure 6: Console login screen

## 4.2 Default users

The Nova Bonita Console has three default users types:

**root :**

This user has the rights to manage the console look and feel, he can also manage :

- The navigations and pages of the console
- The languages setting
- The users, groups and memberships
- The registry of all the Nova Bonita Console applications

Finally, this user has by default the two profiles User and Operator, so he access to all the associated applications.

**admin :**

This user has the Operator profile so he access to the functionalities described in Chapter 1.2

**james, john and jack:**

These users have the User profile so they access to the functionalities described in Chapter 1.1

## 4.3 Console frames description

After logging in, the Nova Bonita Console is available in the main frame, of your browser

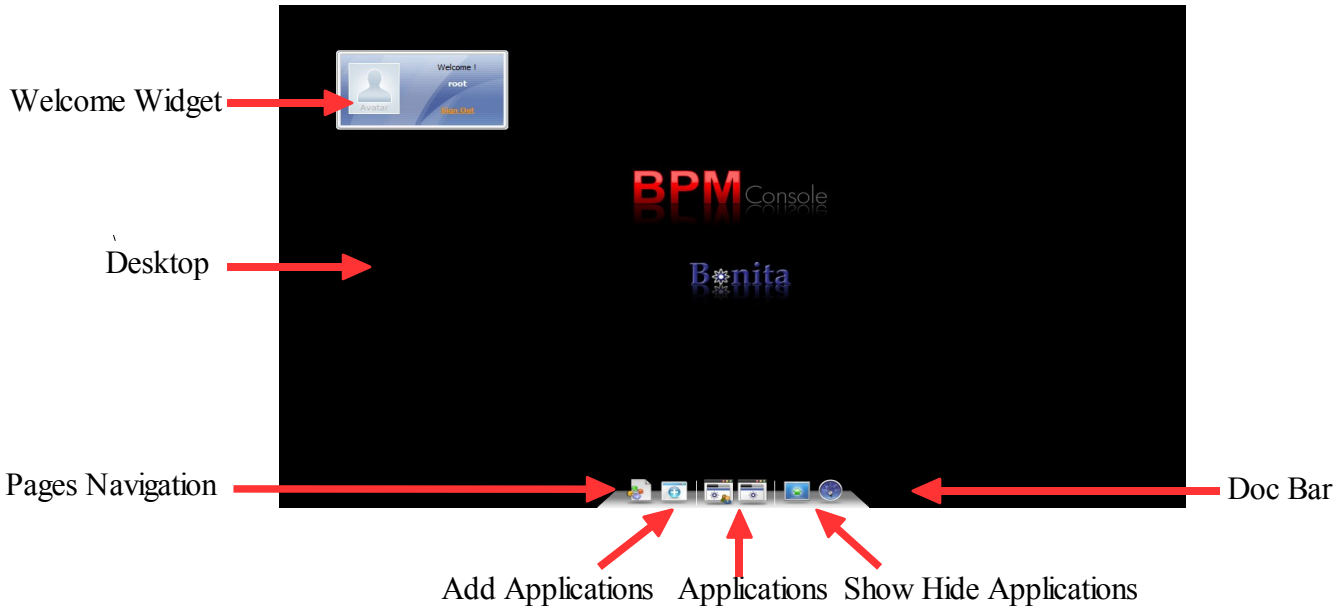



figure 7: Console Description

**Desktop :**

The Desktop is the workspace in which BPM and others applications (aka portlets) can be added, removed, configured... like in an OS

**Doc Bar :**

The Doc Bar is used to access to the available applications for a particular user type

**Applications :**

The Application allow to perform the business functionalities like the process management ...etc. Each application is independent from the others. It looks like a simple window.

**Pages Navigation Button :**

This button allow to display the list of the Console navigations and to navigate between the different pages.

**Add Applications Button :**

This button allow to add an application/widget in the Doc Bar/Desktop.

**Show / Hide Applications :**

This button allow to hide all the Applications / Widgets displayed on the desktop. A second click allow to display them again.

**Welcome Widget :**

The welcome widget display the logged user's name and allow to logout from the Nova Bonita Console.

## 4.4 Application's graphical organization description

Each Application is organized like this :

- **Main Tools :** The main banner that allow to access the main tools.
- **Tabulations :** A set of tabulation for the tools in use. A lot of tabulations can be opened at the same time.
- **Work Area :** The tools display area.

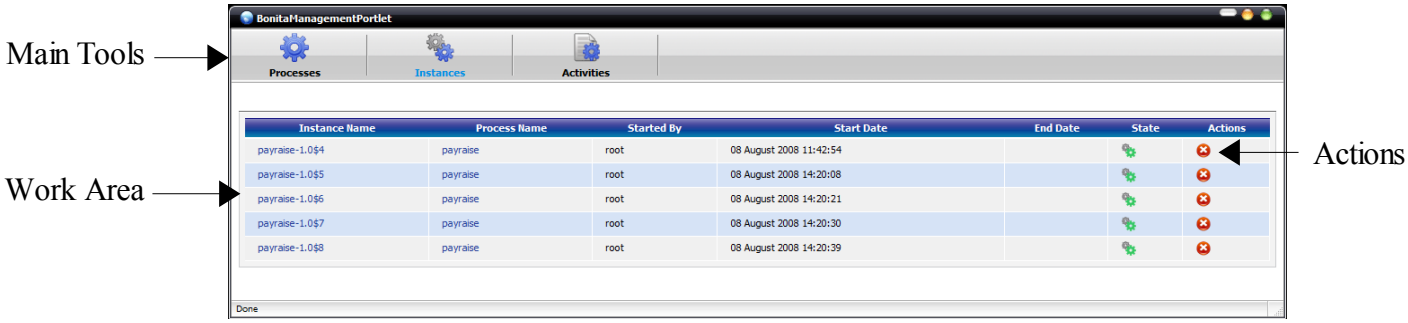



figure 8: Application Description (1/3)

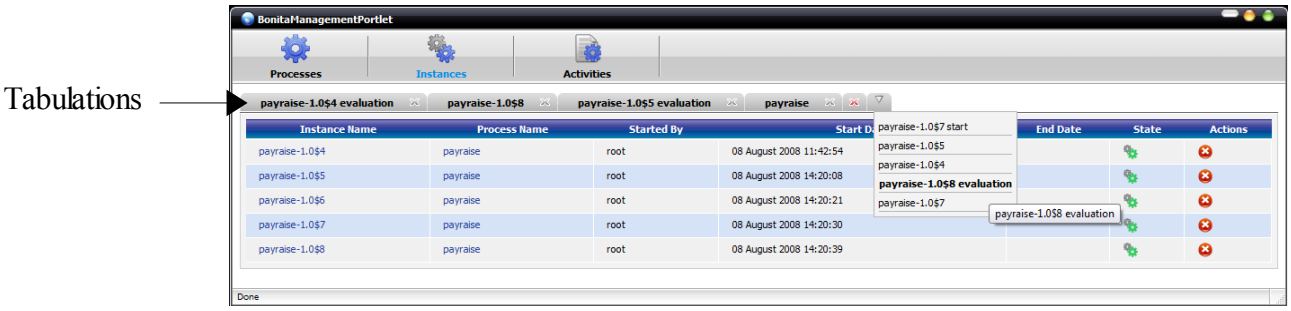



figure 9: Application Description (2/3)

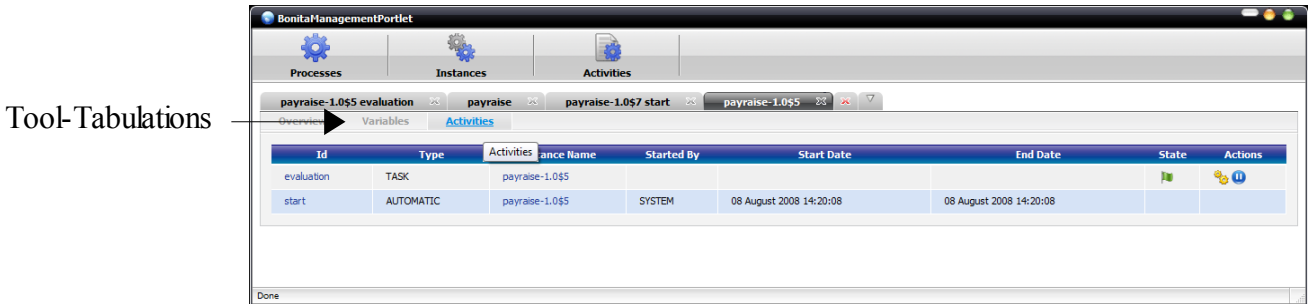



figure 10: Application Description (3/3)

# Chapter 5. Accessing and Creating Processes

## 5.1 Access the Workflow Process List

Select the application Users WorkList in the **Doc Bar**, then select the "To Do List" tabulation

in the **Main Tools** to display the list of all the remaining tasks to be performed and all the accessible processes that can be launched.

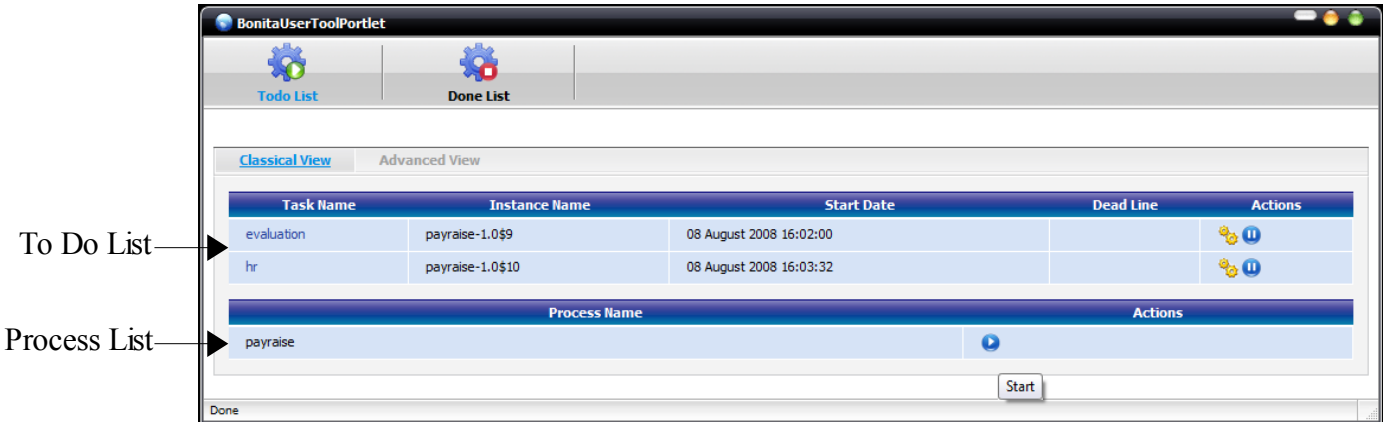



figure 11: Access Workflow Process List

## 5.2 Create a new Instance of a Bonita Process

- Access the workflow process list (see section 4.1).

-To create a new instance of the process, click on the Start button.

- If the start of the process is manual, a form is displayed.

- Fill in the form and click on the submit button as shown in the example below.

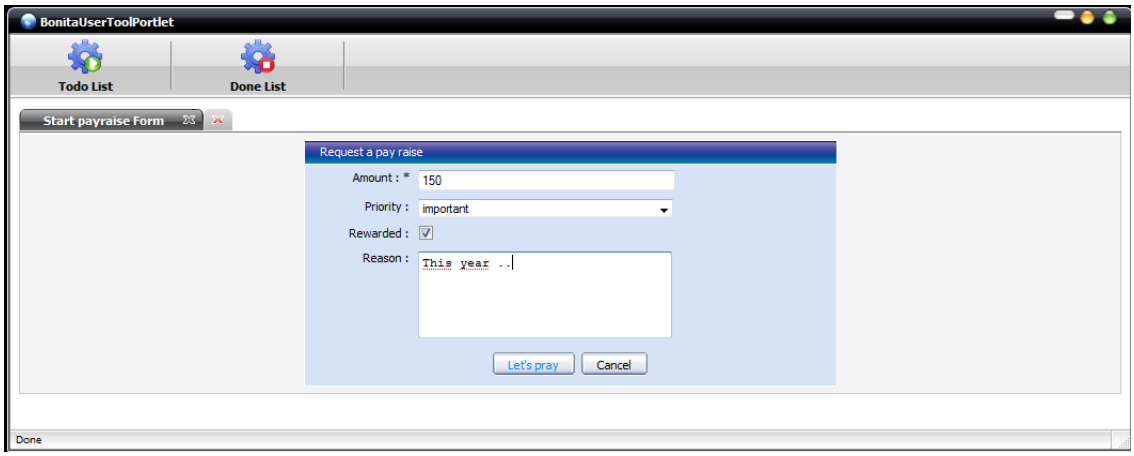



figure 12: Creation of a new instance form example

# Chapter 6. Access To Do / Done Tasks

## 6.1 Consult the To Do Tasks List

- Select the application User Worklist in the **Doc Bar**, then select the "ToDo List" tabulation

in the **Main Tools** to display the list of all the To Do tasks remaining to be performed.

- Two different views are possible for the To Do list :

**Classical view :** This view display the To Do tasks list in a simple grid mode.

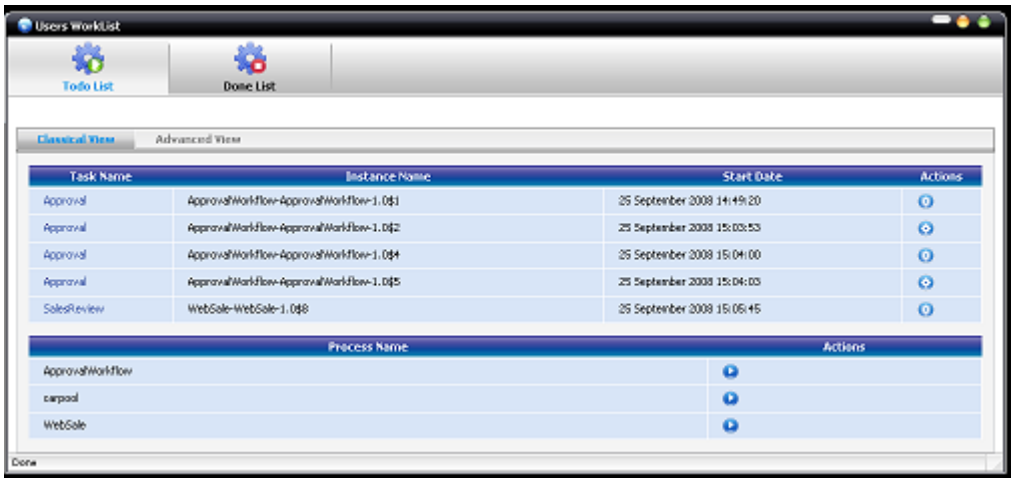



figure 13: Classical view of the To Do List

**Advanced view :** This view display the users tasks list sorted by process and activity names. Once clicking in each activity name the application will list the available activities instances corresponding to a particular activity name.

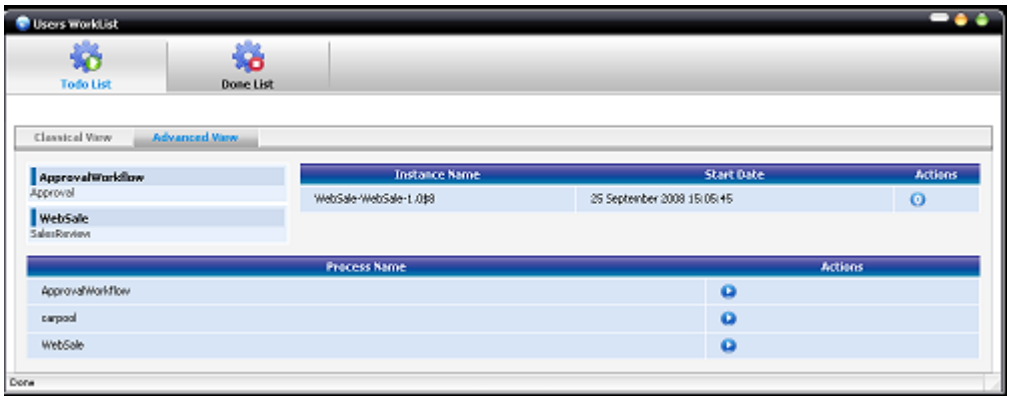



figure 14: Advanced view for the To Do List

- The following parameters are displayed for each task :

- **Task name :** The task name
- **Instance name :** The associated instance name
- **Start Date :** The start date of the task

- A detailed view of a task is available by clicking on the name of the task.

A description text can be displayed if exist.

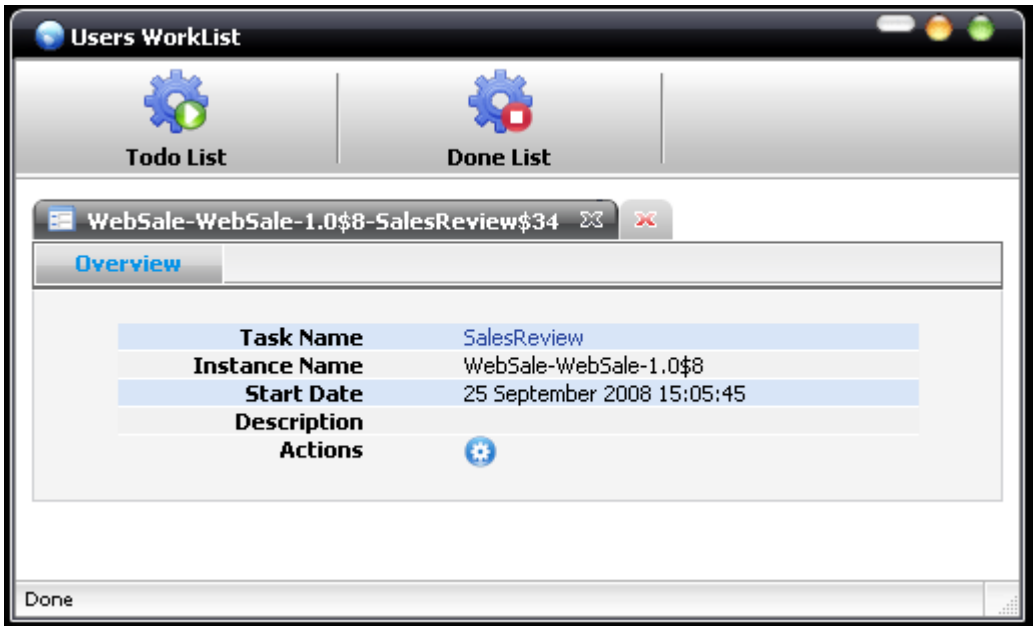



figure 15: Detailed view of Task

## 6.2 Perform / Pause / Resume a Task

### 6.2.1 Perform

- Go to the To Do tasks list (see section 5.1).

- To perform a task, click on the button (perform) in the Actions field of the line corresponding to the task you want to perform (as shown above on the example).

- If this task has properties to be set or read by the user, a form is displayed.

- Fill in the form, then click on the "submit" button as illustrated below.

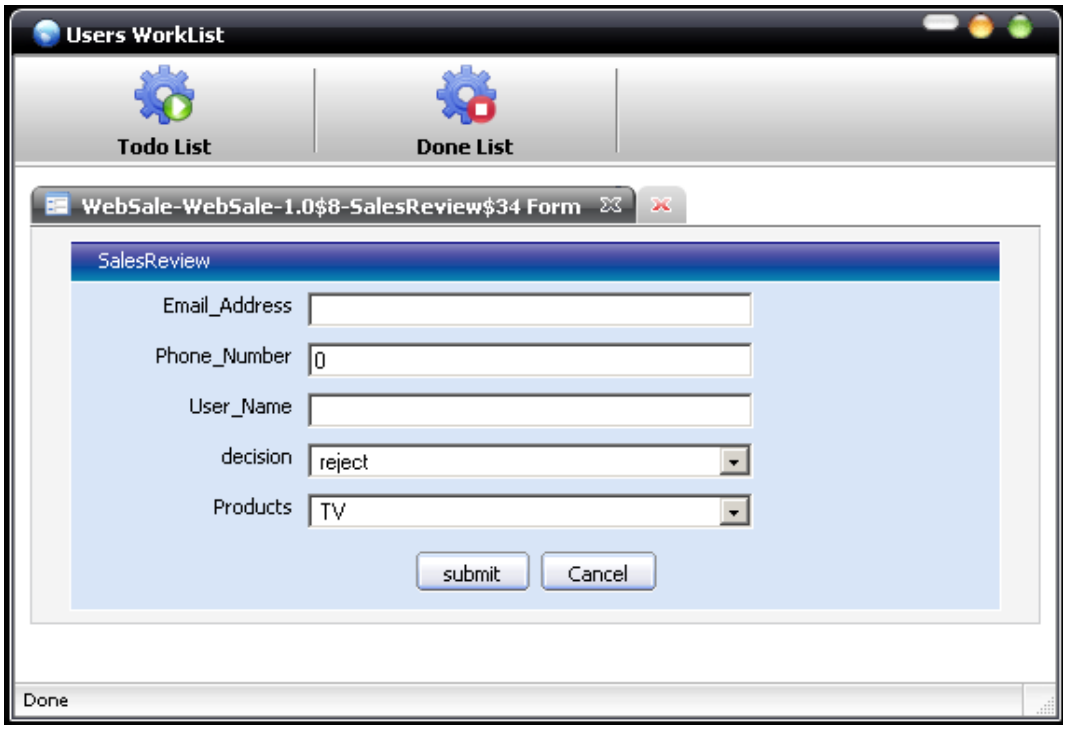



figure 16: Perform a task

### 6.2.2 Suspend

- Go to the To Do tasks list (see section 5.1).

- To suspend a task, click on the button at the end of the line of the task you want to suspend.

### 6.2.3 Resume

- Access the To Do tasks list (see section 5.1).

- To resume a suspended task, click on the button at the end of the line of the task you want to resume (as shown above on the example). Only suspended tasks can be resumed !.

## 6.3 Consult the Done Tasks List

Select the application Users WorkList in the **Doc Bar**, then select the "Done List" tabulation

in the **Main Tools** to display the list tasks that has already been executed.

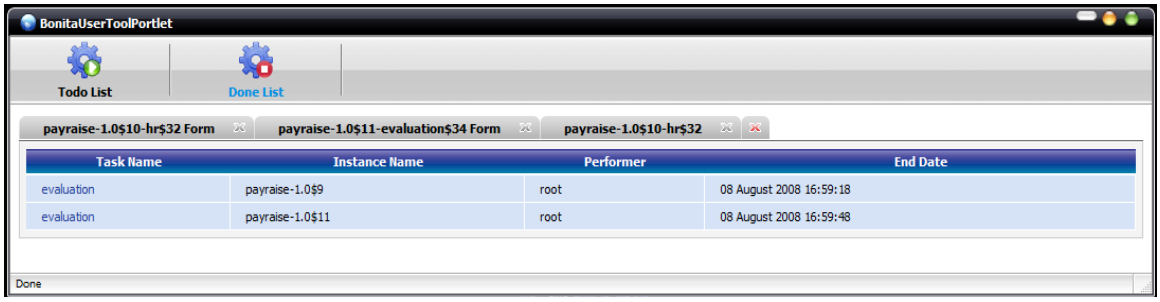



figure 17: The Done Tasks List

# Chapter 7. Managing Process Models

Those features are only accessible by Operators users.

## 7.1 Access the Process Model List

- Select the application BPM Management in the **Doc Bar**, then select the Processes tabulation

in the **Main Tools** to display the list of all the accessible processes.

- The following parameters are displayed for each process model line :

- **Process name :** The name of the process
- **Version :** The version number of the process
- **Description :** Some description about the process

- The following actions are possible for each process model :

- starting the process
- removing all the running instances of this process model
- deleting a process
- undeploying a process

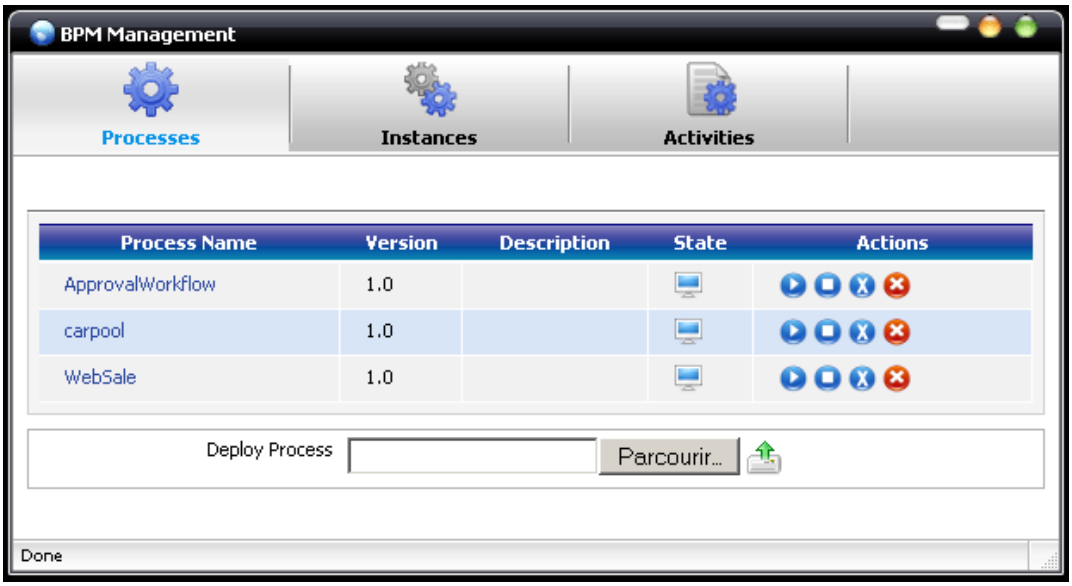



figure 18: Process Model List

-A detailed view of a process model is available by clicking on the name of the process.

figure 19: Detailed view of a process model

-Associated instances of the process model displayed in the detailed view are available by clicking on "Instances" sub-tabulation.

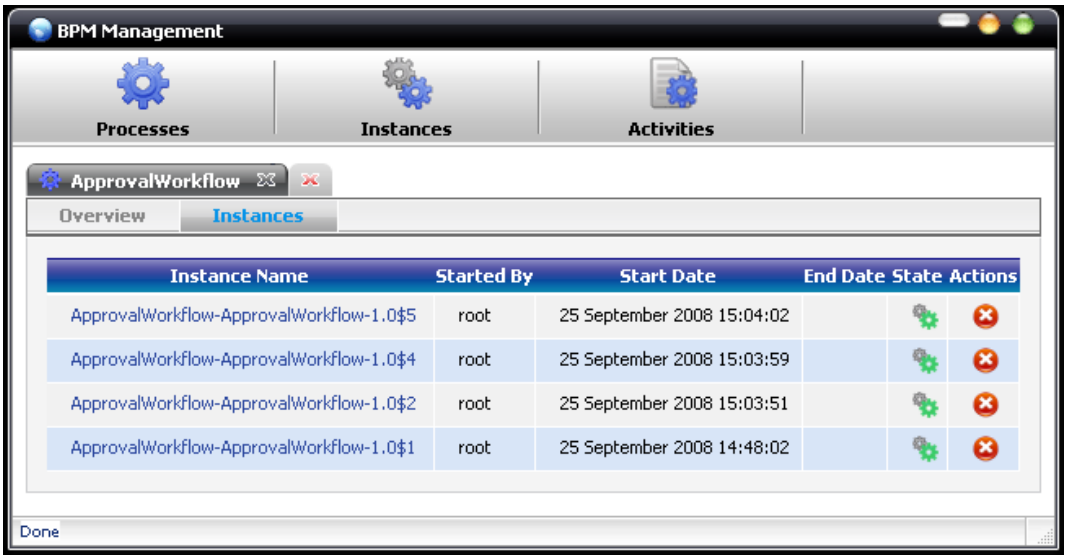



figure 20: Consult the instances list for a given process

## 7.2 Deploy / Undeploy / Delete Processes Models

**Go to** the process models list (see section 6.1).

**Deploy :**

To deploy a process model, click on the browse button and select the path to access your process model, then click on the button (upload) .

After deploying operation, the process models list is refreshed with the new imported process model.

**Undeploy :**

To undeploy a process model, click on the button for the process you want to undeploy. You can also open the detailed view of the process model, then click on the undeploy button in the Actions field.

Note that the undeploying action will keep all historic data in your system.

**Delete :**

To delete a process model, click on button at the end line of the process you want to delete. You can also open the detailed view of a process model, then click on the delete button in the Actions field.

## 7.3 Start Process Models

Access the workflow process list (see section 6.1).

To create a new instance of the process, click on the button (Start button).

If the start of the process is manual, a form is displayed.

## 7.4 Remove all Instances of a Process Model

Access the workflow process list (see section 6.1).

To remove all the instances of a process, click on the button

# Chapter 8. Managing Instances

## 8.1 Access the Process Instances List

Select the application Bonita Management in the **Doc Bar**, then select the Instances tabulation

in the **Main Tools** to display the list of all the running instances.

The following parameters are displayed for each process instance :

- **Instance name :** The name of the instance (click to display the detailed view of the instance)
- **Process name :** The name of the associated process model (click to display the detailed view of the process)
- **Started By :** The name of the user that started the instance
- **Start Date :** The date in which the instance was started
- **End Date :** The finish date of the instance
- **State :** The state of the instance

In the following table we present the different states of instances.

| State | Meaning |
|---|---|
| | Started : The instance is in started state |
| | Finished : The instance is in the finished state |
| | Initial : The instance is in the initial state |

The following actions are possible for each process instance line :

- Delete the instance

- It is possible to consult a detailed view of a process instance by clicking on the name of the instance.

figure 21: Detailed View of an Instance

## 8.2 Consult / Edit the variables of an instance

- Access the instances list (see section 7.1).

- Click on the instance name to display the detailed view of the instance.

- Click on the "Variables" sub-tabulation to display all the associated variables of the instance. You can also EDIT those variables by clicking on the button and filling the new value in the variable edition popup.

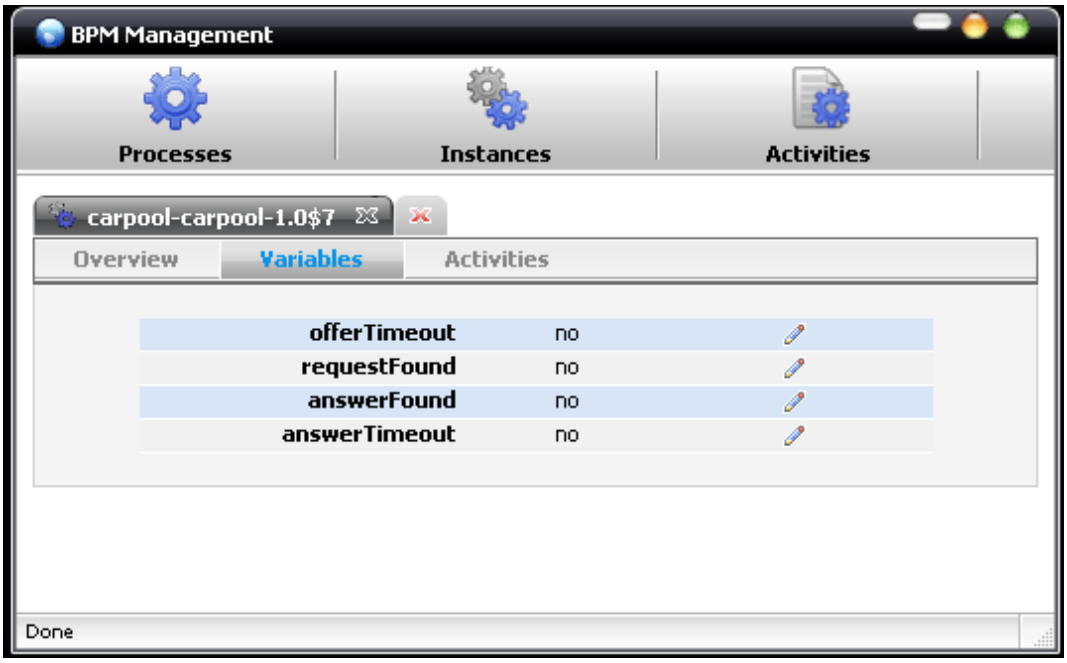



figure 22: Consult the variables of an instance

## 8.3 Access the activities list of an instance

- Go to the instances list (see section 7.1).

- Click on the instance name to display the detailed view of the instance.

- Click on the "Activities" sub-tabulation to display all the associated activities of the instance.

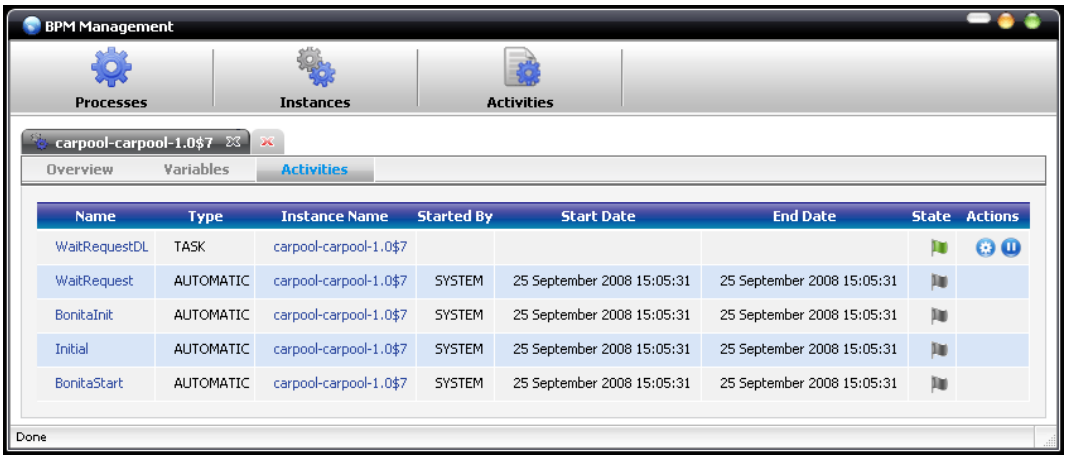
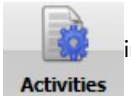



figure 23: Consult Activities of a given instance

# Chapter 9. Managing Activities

## 9.1 Access the Activities List

Select the application Bonita Management in the **Doc Bar**, then select the Activities tabulation 

in the **Main Tabulation** to display the list of all the Activities.

The following parameters are displayed for each activity line :

- **Id :** The id of the activity (click to display the detailed view of the activity)
- **Type :** The type of the activity
  - AUTOMATIC : Automatic execution of the activity
  - TASK : manual activity that requires human interaction
  - SUBFLOW : subprocess activity that creates a subprocess
- **Instance name :** The name of the instance that triggered the activity (click on to display the detailed view of the instance)
- **Started By :** The name of the user that started the activity
- **Start Date :** The lunch date of the activity
- **End Date :** The finish date of the activity
- **State :** The state of the activity

In the following table you will find the different states of the activities.

| State | Meaning |
|---|---|
| | Ready : The activity is ready to be started |
| | Initial : The activity is in the initial state |
| | Executing : The activity is in execution |
| | Finished : The activity has already been finished |
| | Suspended : The activity has been suspended |

- A detailed view of an activity is available by clicking on the name of the activity.

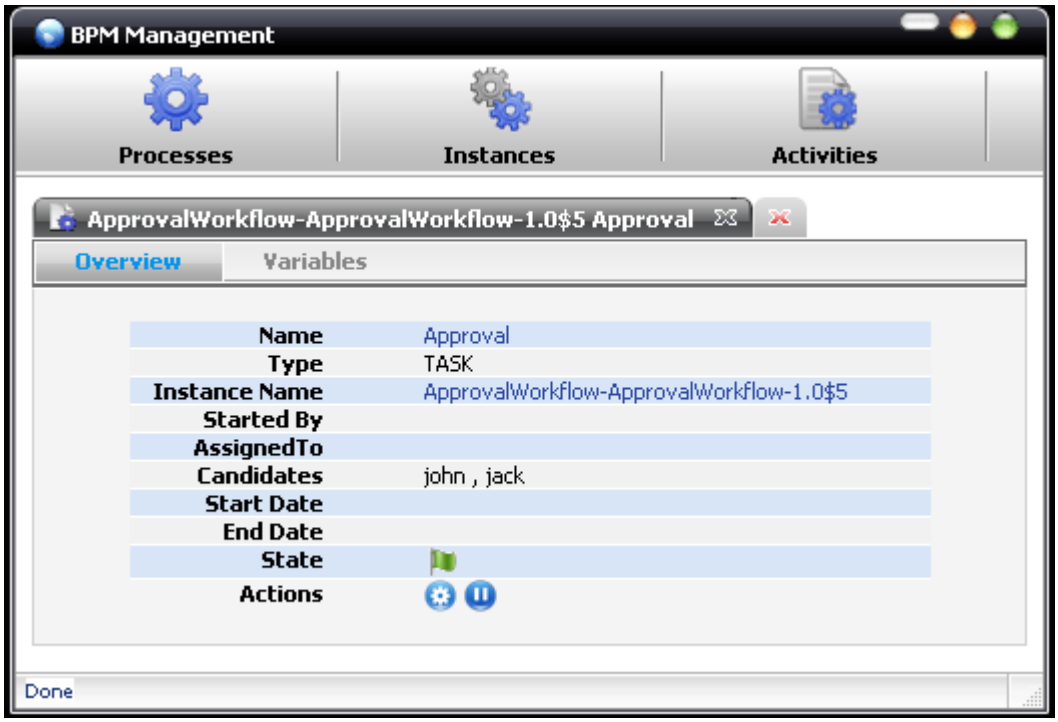



figure 24: Detailed view of an activity

## 9.2 Start an Activity

- Go to the activities list (see section 8.1).

- To perform an activity, click on the button in the Actions field of the line corresponding to the activity you want to start (as shown above on the example).

- If this activity has properties to be set or read by the user, a form is displayed.

- Fill in the form, then click on the "submit" button.

## 9.3 Suspend an Activity

-Access the activities list (see section 8.1).

- To suspend an activity, click on the button at the end of the line of the activity you want to suspend (as shown above on the example).

## 9.4 Resume an Activity

- Access the activities list (see section 8.1).

- To resume a suspended activity , click again on the button at the end of the line of the activity you want to resume.

## 9.5 Access the Variables List of an Activity

-Access the activities list (see section 8.1).

-Click on the instance name to display the detailed view of the activity.

-Click on the "Variables" sub-tabulation to display all the associated variables of the activity.

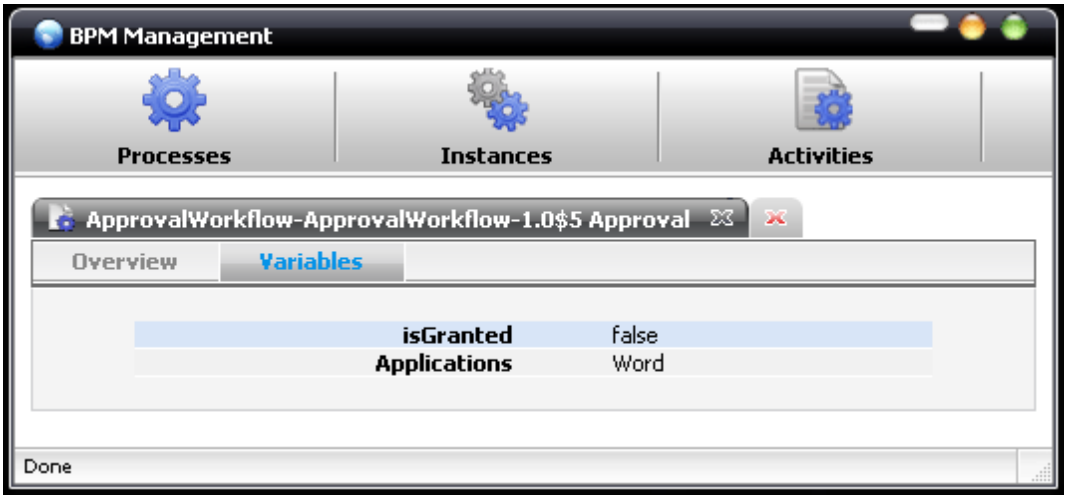



figure 25: Access the variables of an activity

# Chapter 10. Managing users, groups and memberships

Those functionalities can be done only by the **root** user.

This user is allowed to view the left side workspace bar that gives access to administration features of the Console.

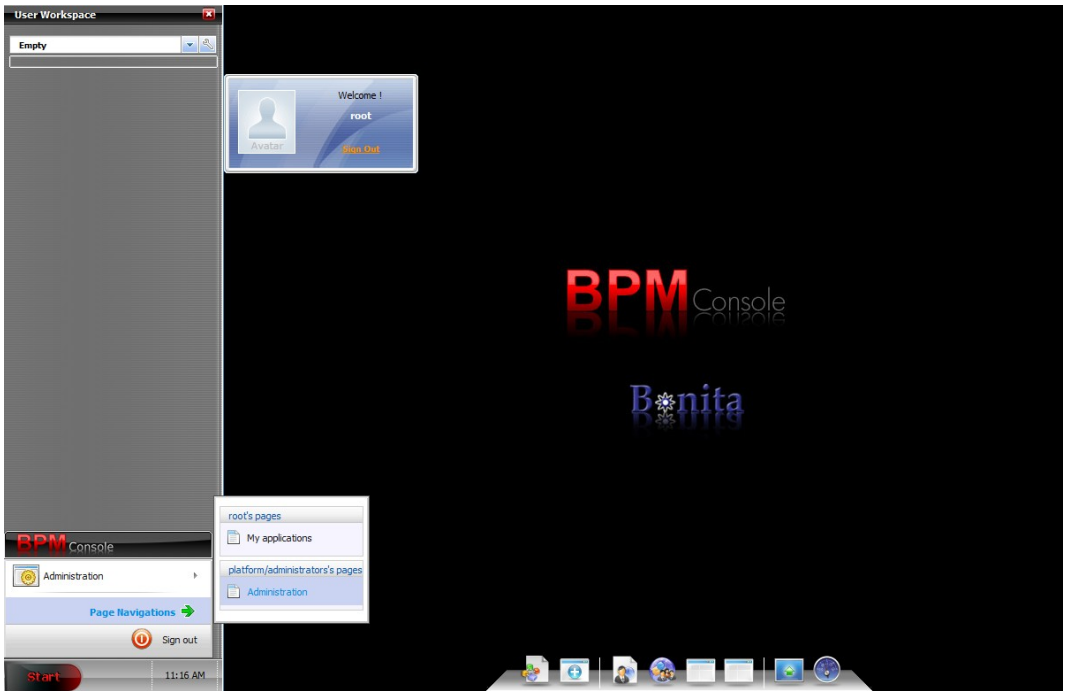



figure 26: Access Administration page

## 10.1 Add a new user to the Nova Bonita Console

Then open the "New Account" application, and fill the form with the informations of the new user and finally click on "Save " button to validate.

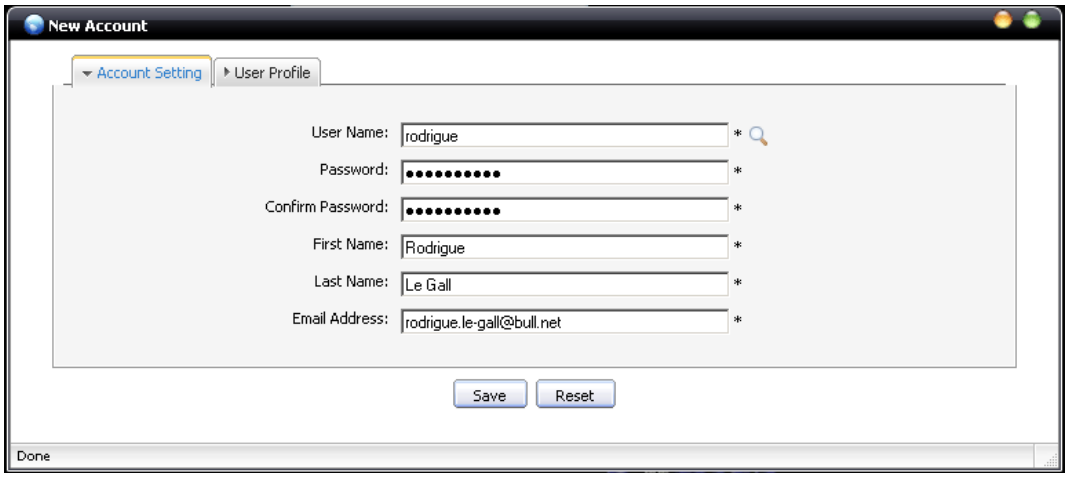



figure 27: Add new user to the Nova Bonita Console

## 10.2 Set roles/permission of a user to access to the console

- Go to the Administration page with the "**Pages Navigation**" button or the root navigation as shown in the figure 36.

- Then open the "**Community Management**" application.

- Click on "**Group Management**" tabulation.

- Select the group **Platform > Console > Bonita**

- After that just add the user you want to give access to the console with at least the membership "**user**". And repeat the operation if you want that user has the role "**operator**".

Example : Setting the user "rodrigue" as operator of the console means that we need to put it in the group Bonita with 2 memberships : user and operator.

Step 1 : Select the right group

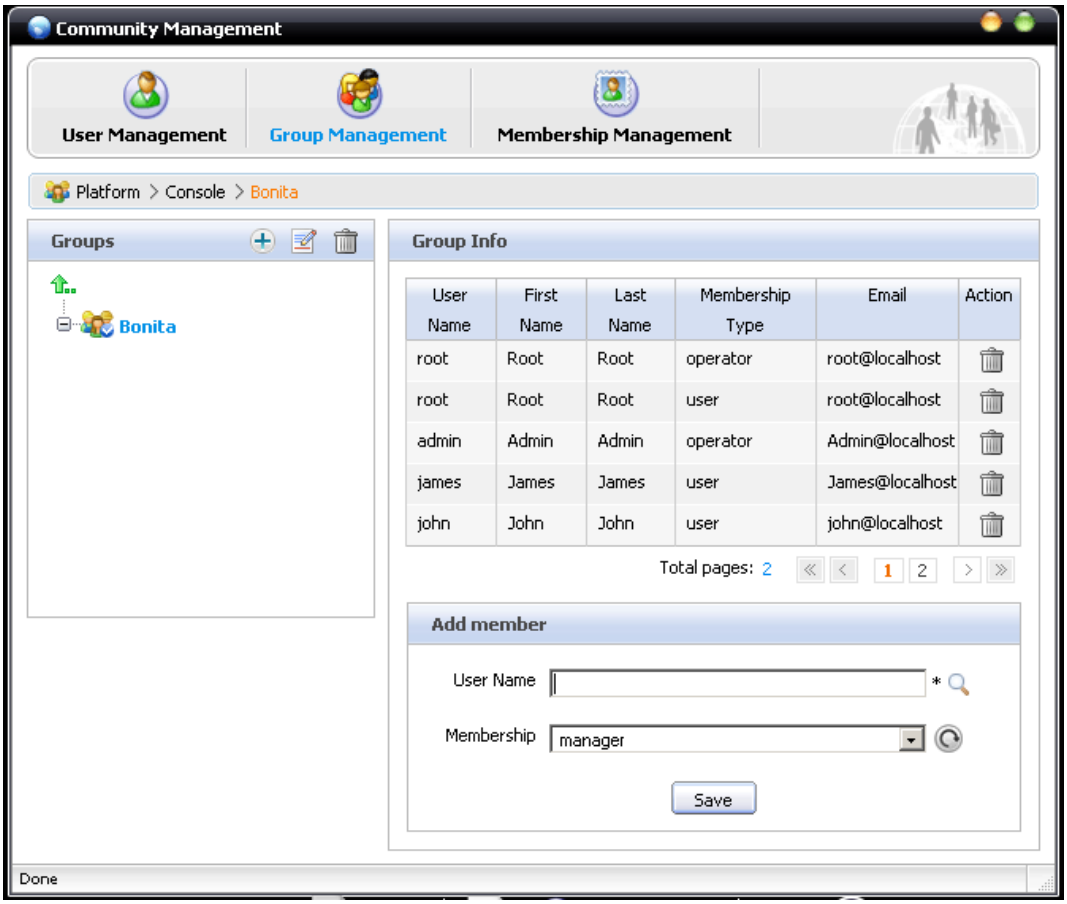



figure 28: Select the Bonita group

Step 2: fill the Add member form with user name rodrigue and Membership user and click on Save

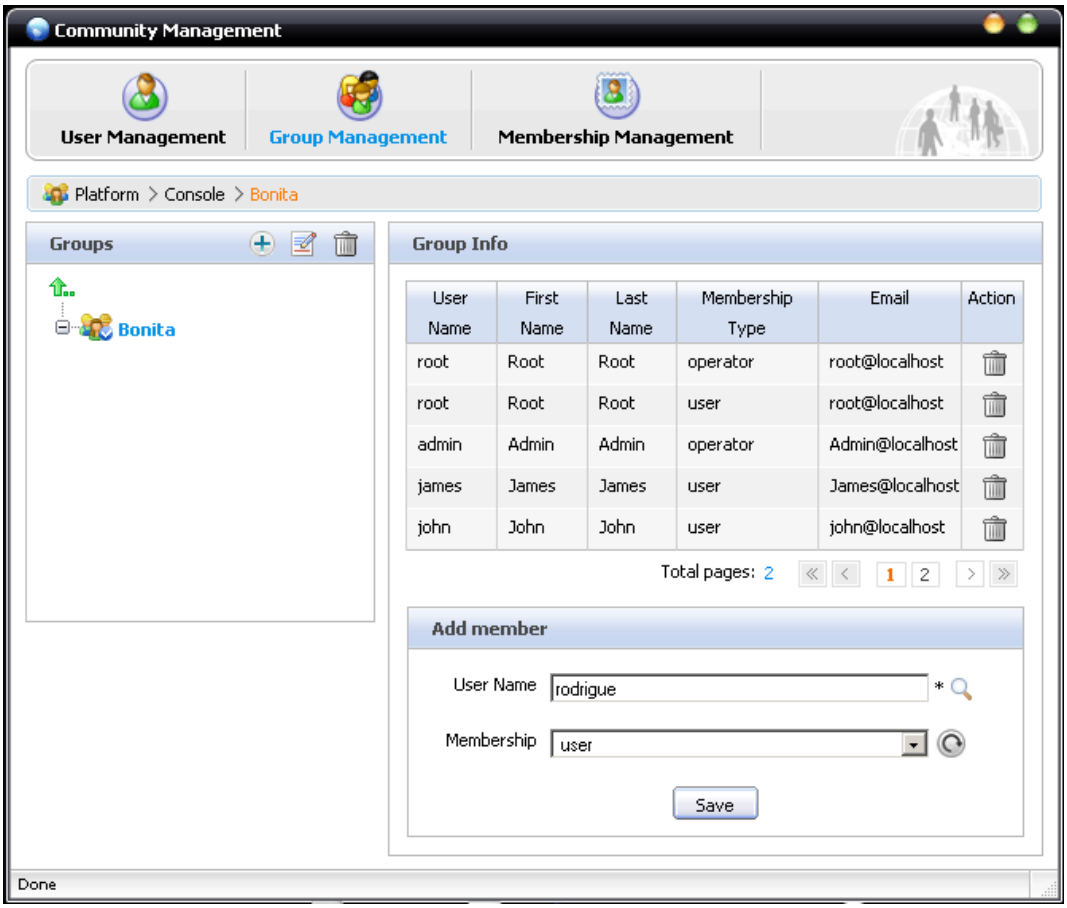



figure 29: fill the Add member form to add as user

Step 3 : fill the Add member form with user name rodrigue and Membership operator and click on Save

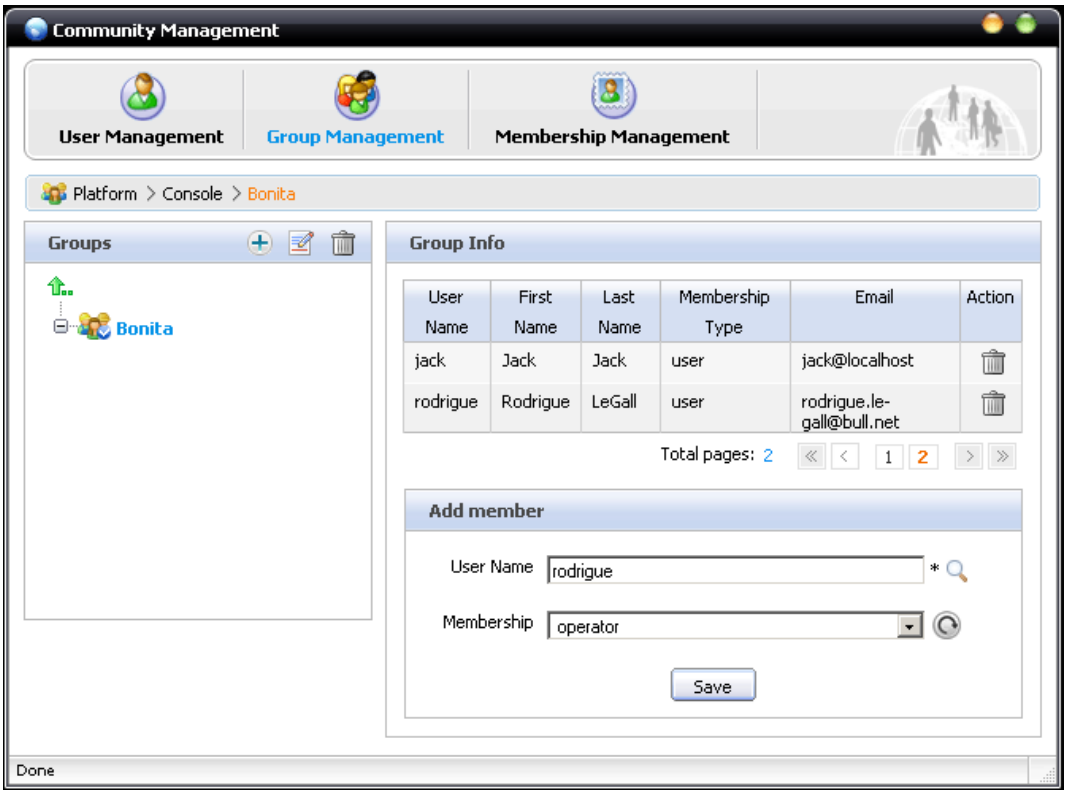



figure 30: fill the Add member form to add as operator

The user is now an operator of the Nova Bonita Console.

# Chapter 11. Forms customization

## 11.1 Overview

The Bonita console is built with an automated form generator. This functionality is useful during the conception and testing phases of your processes. The major inconvenient of the automated form generation is that generated forms are not user friendly. In order to solve this problem, the form generator is customizable via few configuration files.

To customize the forms of a Bpm process, you just need to write a description file named **forms.xml** . Each web form can be internationalized by means of a properties files. All those files must be located in the root directory of the bar (Business Archive) file of your process.

This chapter will explain how to write **forms.xml** file and the internationalized property files.

## 11.2 Forms.xml syntax

### 11.2.1 Overview

The forms.xml syntax is based on an xml syntax initially derived from the xFormsyntax.

The form generator suppose that you have only one form by manual activity in your process.

Here is the abstract of all tags you need to know to customize your forms

```
<forms>
      <form>
            <activity/>
            <resource-bundle/>
            <customized-view/>
            <variable>
                  <validator>
                        <property/>
                  </validator>
            </variable>
            <message/>
      </form>
</forms>
```

### 11.2.2 Tag list

#### 11.2.2.1 <forms/>

**Description :** top level tag of the forms description.

**Mandatory** : true

**Properties :** none

**Childs :** <form/>

#### 11.2.2.2 <form/>

**Description** : include the description of a specific activity form

**Mandatory** : true

**Properties** : none

**Childs** : <activity/>, <resource-bundle/>, <customized-view/>, <variable/>, <message/>

#### 11.2.2.3 <activity/>

**Description** : include the id of the activity in your process model. If you want to write a form to start a process, you need to enter an empty value.

**Mandatory** : true

**Properties** : none

**Childs** : none

**Examples** :

Write a form for the activity "validation" of your process:

```
<forms>
	<form>
		<activity>validation</activity>
		...
	</form>
	....
</forms>
```

Write a form to start your process :

```
<forms>
	<form>
		<activity></activity>
		...
	</form>
	....
</forms>
```

#### 11.2.2.4 <resource-bundle/>

**Description** : the resource bundle includes the translation of each variable names into human friendly language. The value refere to a .properties file : <resource-bundle>.properties

**Mandatory** : false

**Properties** : none

**Childs** : none

**Examples** :

Write a form for the activity "validation" that will use the following files for translation in english, french and spanish : validation-i18n.properties , validation.i18n_en.properties, validation.i18n_fr.properties, validation.i18n_es.properties

```
      <forms>
            <form>
                  <activity>validation</activity>
                  <resource-bundle>validation.i18n</resource-
bundle>
                  ...
            </form>
            ....
      </forms>
```

### 11.2.2.5 <customized-view/>

**Description** : specify the local path of a template file that will be used for the facing of your form. This template have to be written in groovy.

**Mandatory** : false

**Properties** : none

**Childs** : none

### 11.2.2.6 <variable/>

**Description** : define a variable of the process that will be used in the form.

**Mandatory** : false

**Properties** :

- name

  **Description** : the id of the variable in the process

  **Mandatory** : true

  **Possible values** : none

  **Default value** : none

  **Example** :

  Write a form for an activity "validation" that use a variable "comment".

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="comment"/>
      </form>
      ....
</forms>
```

- component

**Description** : the type of widget you want to use in your form to interact with your variable

**Mandatory** : false

**Possible values** : text, textarea, select, checkbox, radiobox, wysiwyg, date, date-time

**Default value** : text

**Example** :

Write a form for an activity "validation" that use a variable "grant" with a checkbox.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="grant" component="checkbox"/>
      </form>
      ....
</forms>
```

- editable

  **Description** : specify if the variable is editable or not

  **Mandatory** : true

  **Possible values** : true, false

  **Default value** : true

  **Example** :

  Write a form for an activity "validation" that use a variable "explaination" that should only be printed and not modified.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="explaination" editable="false"/>
      </form>
      ....
</forms>
```

- mandatory

  **Description** : specify if the variable must be filled or not

  **Mandatory** : false

  **Possible value** : true, false

  **Default value** : false

  **Example** :

Write a form for an activity "validation" that use a variable "amount" that must be filled.

```
<forms>
	<form>
		<activity>validation</activity>
		<variable name="amount" mandatory="true"/>
	</form>
	....
</forms>
```

**Childs** : <validator/>

### 11.2.2.7 <validator/>

**Description** : add a validator for the data of the parent variable. If the validator failed, an error is show in the form.

**Mandatory** : false

**Properties** :

- name

  **Description** : the name of the validator

  **Mandatory** : true

  **Possible values** : DateTime, EmailAddress, Expression, Number, NumberInRange, PositiveNumber, SpecialCharacter, StringLength, Float

  **Default value** : none

**Childs** : <property/>

**Example** :

Wite a form for activity "validation" with a variable "amount" that must be a Float and mandatory.

```
<forms>
	<form>
		<activity>validation</activity>
		<variable name="amount" mandatory="true">
			<validator name="Float"/>
		</variable>
	</form>
	....
</forms>
```

### 11.2.2.8 <property/>

**Description** : define a property needed by a validator

**Mandatory** : false

**Properties** :

- name

  **Description** : the name of the property

  **Mandatory** : true

  **Possible values** : none

  **Default value** : none

- value

  **Description** : the value of the property

  **Mandatory** : true

  **Possible values** : none

  **Default value** : none

**Childs** : none

**Example** :

Write a form for activity "validation" with a variable "amount" that must be between 100 and 10000.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="amount">
                  <validator name="NumberInRange">
                        <property name="min" value="100"/>
                        <property name="max" value="10000"/>
                  </validator>
            </variable>
      </form>
      ....
</forms>
```

### 11.2.3 Data Validators list

#### 11.2.3.1 DateTime

**Description** : Valides that the data is a date

**Properties** : none

**Example** :

Write a form for an activity "schedule" with a variable meeting

```
<forms>
      <form>
            <activity>schedule</activity>
            <variable name="meetingDate" component="date-time">
                  <validator name="DateTime"/>
            </variable>
      </form>
      ....
</forms>
```

#### 11.2.3.2 EmailAddress

**Description** : Validates that the data is an email address

**Properties** : none

**Example** :

Write a form for an activity "signup" with a variable "email".

```
<forms>
      <form>
            <activity>signup</activity>
            <variable name="email" mandatory="true">
                  <validator name="EmailAddress"/>
            </variable>
      </form>
      ....
</forms>
```

#### 11.2.3.3 Expression

**Description** : Validates that the data matches one regular expression.

See http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#sum for the regular expression format.

**Properties** :

- expression

  **Description** : this is the regular expression that is conform to the format describe here http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#sum

  **Mandatory** : true

**Example** :

Write a form for an activity "signup" with a variable "email" validated with your own email pattern.

```
<forms>
      <form>
            <activity>signup</activity>
            <variable name="email" mandatory="true">
                  <validator name="Expression">
                        <property name="expression" value="^[A-
Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$"/>
                  </validator>
            </variable>
      </form>
      ....
</forms>
```

### 11.2.3.4 Number

**Description** : Validates that the data is a number (interger).

**Properties** : none

**Example** :

Wite a form for activity "validation" with a variable "amount" that must be a number.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="amount">
                  <validator name="Number"/>
            </variable>
      </form>
      ....
</forms>
```

### 11.2.3.5 NumberInRange

**Description** : Validates that the data is a number in a specified range

**Properties** :

- min

  **Description** : the minimum of the range

  **Mandatory** : true

- max

  **Description** : the maximum of the range

  **Mandatory** : true

**Example** :

Wite a form for activity "validation" with a variable "amount" that must be between 100 and 10000.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="amount">
                  <validator name="NumberInRange">
                        <property name="min" value="100"/>
                        <property name="max" value="10000"/>
                  </validator>
            </variable>
      </form>
      ....
</forms>
```

#### 11.2.3.6 PositiveNumber

**Description** : Validates that the data is a positive number

**Properties** : none

**Example** :

Wite a form for activity "validation" with a variable "amount" that must be a positive number.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="amount">
                  <validator name="PositiveNumber"/>
            </variable>
      </form>
      ....
</forms>
```

#### 11.2.3.7 SpecialCharacter

**Description** : Validates that the data only contains letters, digits, '-', '_' or space

**Properties** : none

**Example** :

Wite a form for activity "validation" with a variable "id" that must not contains any special characters

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="id">
                  <validator name="SpecialCharacter"/>
            </variable>
      </form>
      ....
</forms>
```

#### 11.2.3.8 StringLength

**Description** : Validates that the data is a string with a length in a range

**Properties** :

- min

  **Description** : the minimum of the range

  **Mandatory** : false

  **Default value** : 0

- max

  **Description** : the maximum of the range

  **Mandatory** : true

**Example** :

Wite a form for activity "signup" with a variable "login" with a length between 4 and 15.

```
<forms>
      <form>
            <activity>signup</activity>
            <variable name="login">
                  <validator name="StringLength">
                        <property name="min" value="4"/>
                        <property name="max" value="15"/>
                  </validator>
            </variable>
      </form>
      ....
</forms>
```

### 11.2.3.9 Float

**Description** : Validates that the data is a float (ie: 100.45)

**Properties** : none

**Example** :

Wite a form for activity "validation" with a variable "amount" that must be a Float.

```
<forms>
      <form>
            <activity>validation</activity>
            <variable name="amount">
                  <validator name="Float"/>
            </variable>
      </form>
      ....
</forms>
```

## 11.3 Internationalize your forms

### 11.3.1 Overview

To internationalize a form, you need to provide a resource bundle (see **<resource-bundle/>**). The name of the resource bundle corresponds to the principal part of the name of internationalized property file.

Example: the resource bundle for the activity "**validation**" is "**validation.i18n**", then the corresponding files will be **validation.i18n.properties** or **validation.i18n_*<locale>*.properties** where locale is en for english, fr for french, es for spanish, ...

So if you want to internationalize you forms, you need to write a file by language and by activity and refer them in the **forms.xml** file by using **<resource-bundle/>** tag. These files must be included in the root directory of the BAR (Business Archive) file of your process.

### 11.3.2 Syntax

#### 11.3.2.1 Overview

The syntax of the *.properties files are very simple, each line may contains a couple key/value like this : **key=value** .

To comment a line you need to start this line with **#**.

Example :

```
task-name=update the salary system
title=Update Salary

submit=Confirm update

#variable title and label

initiator.label=Employee :
amount-granted.label=Amount to be added :
```

#### 11.3.2.2 Key list

| Use Case | Key |
|---|---|
| Task name | **task-name** |
| Form title | **title** |
| Submit button | **submit** |
| Variable label | *variableId*.**label** |

| | |
|---|---|
| Checkbox value | variableId.**checkbox** |
| Radiobox values | *variableId*.**radiobox-0**<br>*variableId*.**radiobox-1**<br>...<br>*variableId*.**radiobox-*n*** |
| Select values | *variableId*.**select-0.label**<br>*variableId*.**select-0.value**<br>*variableId*.**select-1.label**<br>*variableId*.**select-1.value**<br>...<br>*variableId*.**select-*n*.label**<br>*variableId*.**select-*n*.value** |