Documentation Bonita

Bonita Team ()

- Oct 2007 -

Table of Contents

Introduction iii
1. General information 1
    1.1. Nova Bonita introduction 1
    1.2. Feature list 1
    1.3. Restrictions 1
2. Prerequisites 3
    2.1. Hardware 3
    2.2. Software 3
3. Installation guide 4
    3.1. Installation 4
    3.2. Bonita directory structure 4
4. User guide 6
    4.1. Designing a xpdl process with ProEd 6
    4.2. Configuration environment 6
    4.3. Getting started with Bonita API 6
        4.3.1. XpdlImport interface 7
        4.3.2. User interface 7
        4.3.3. Process interface 7
    4.4. Running the example 7

Introduction

This documentation is targetted to Bonita users. It presents the installation procedure and a small user guide of Nova Bonita Milestone 1.

Chapter 1 , General information describes the new version Bonita v4 called Nova Bonita

Chapter 2 , Prerequisites describes the hardware and software prerequisites

Chapter 3, Installation guide describes how to install the Nova Bonita Milestone 1

Chapter 4, User Guide guides you through the descovery Nova Bonita functionalities.

Chapter 1. General information

1.1. Nova Bonita introduction

Nova Bonita is the name of new version of Bonita v4.

“Nova” technology is based on the “Process Virtual Machine” conceptual model for processes. The Process Virtual Machine defines a generic process engine enabling support for multiple process languages (such BPEL, XPDL…).

On top of that, it leads to a pluggable and embeddable design of process engines that gives modelling freedom to the business analyst. Additionally, it enables the developer to leverage process technology embedded in a Java application.

For more information about the Process Virtual Machine, check Nova Bonita FAQs [http://wiki.bonita.objectweb.org/xwiki/bin/view/Main/FAQ] on the Bonita web site [http://bonita.objectweb.org].

1.2. Feature list

Nova Bonita (aka Bonita v4) is a lightweight workflow/BPM solution that provide XPDL support. Nova Bonita M1 provides partial XPDL support. Hereafter you can find the list of features included in this milestone:

- Support of the basic XPDL 1.0 activities : Join, Split, manual, automatic, route
- Basic support of advanced entities/resources: Hooks, mappers and performer assignments
- First implementation of TaskRepository and Human Task modules
- Only in memory execution
- Transitions conditions basic support based on BeanShell
- Basic User's API: instantiateProject, startActivity, terminateActivity, getTodoList methods
- Partial support of ProEd XPDL designer
- no security and identity service

1.3. Restrictions

This first milestone of Nova Bonita push out an innovative architecture based on a generic and extensible engine, called "The Process Virtual Machine" and a powerful injection technology allowing services pluggability.

Nova Bonita M1 also include basic support for elements defined in the XPDL 1.0 standard. Future milestones will continue to improve the standard coverage and services support, i.e persistence, timers, notifications... Check the roadmap [http://wiki.bonita.objectweb.org/xwiki/bin/view/Main/Roadmap] for more information.

The first milestone does not support the following features:

- user base

  There's no connection to a user base for the definition and the execution of the workflow (then no ldap support). Then arbitrary participants should be added at design step.

- block activity
- subprocess
- process versionning
- iteration
- deadline
- role initiator of a process
- propagation of activity properties
- no connexion between proEd and the workflow engine
- no actions supported (bean shell hooks)
- hook:

  no hook on process instantiate supported

  as there's no transaction yet for execution there's no difference between before and after terminate types
- role mapper:

  no property and ldap type supported (only custom) Enter manually the class name of the hook that exists in the classpath of your application
- performer assignment

  no property type supported (only callback)
- transition conditions are only based on process properties (propagated properties could not yet been taken in account)

Chapter 2. Prerequisites

2.1. Hardware

A 1GHz processor is recommended, with a minimum of 512 Mb of RAM. Windows users can avoid swap file adjustments and get improved performance by using 1Gb or more of RAM

2.2. Software

- Nova Bonita requires Java Development Kit (JDK) 1.5 (also called JDK 5.0) but also runs with with next release.

  The JDK software can be downloaded from http://java.sun.com/j2se/1.5.0

- Nova Bonita requires Apache Ant 1.6.5 or higher

  It can be downloaded from http://ant.apache.org

Chapter 3. Installation guide

3.1. Installation

Unzip the Bonita distribution package.

```
>unzip bonita-4.0.M1.zip
```

A new directory `bonita-4.0.M1` will be created with the following structure:

```
README
build.xml
License.txt
conf/
doc/
examples/
lib/
```

3.2. Bonita directory structure

Hereafter is detailled the structure of Bonita installation. The installation directory contains the following structure :

```
README
build.xml
License.txt
conf/
doc/
examples/
lib/
```

Let's present those items :

- README

  This file gives the basic information related to Nova Bonita

- build.xml

  This file is an ant file that provides tasks to run both unit tests and examples (detailled command are given in following sections).

- License.txt

  The license of Nova Bonita. Bonita is released under the LGPL license.

- conf/

  This directory contains the default configuration for Nova Bonita. This xml file, called environment.xml contains the services and objects used as default in Nova Bonita

- doc/

  This directory contains the documentation of Nova Bonita. It contains 2 directories :

  - html/

    For HTML documentation

  - pdf/

    For PDF documentation

- examples/

This directory contains an example provided with Nova Bonita package. The example is under the directory

- BonitaSimpleProjectEver

  This is simple Approval Workflow process

- lib/

  This directory contains the libraries used in Nova Bonita.

Chapter 4. User guide

This chapter describes how to use Nova Bonita M1. This chapter follows steps required to write a Bonita based sample/application which are:

- creating Bonita process definition
- importing the process defintion
- writing application/sample using Bonita interfaces

4.1. Designing a xpdl process with ProEd

If you are already a user of Bonita v3, you probably remember that processes could be created either through a java api or through a graphical editor : ProEd. As java api to build Bonita v4 processes is not yet developped, processes should be designed under proed (see restrictions below) and then imported as xpdl files. For that, use the stand alone version of the process editor that can be downloaded on Bonita download [http://forge.objectweb.org/project/showfiles.php?group_id=56&release_id=302/] page.

Please, refer to the online [http://wiki.bonita.objectweb.org/xwiki/bin/download/Main/Documentation/Bonita_DevelopmentGuide.pdf] documentation to use proEd.

As final result proEd saves the description of the designed process as an xpdl file that should be imported as described in the following section.

4.2. Configuration environment

The PVM includes a framework to allow the injection of services and objets that will be leveraged during the workflow definition and execution. Objects and services required in Bonita are defined through an XML file. A dedicated parser and wiring framework in the PVM is in charge of creating those objects.

A default environment file (environment.xml) is provided in the installed package.

Currently, following objects are required for the execution environment :

- deployer
- instanceRepository
- taskRepository
- processRepository
- processRepository
- var.mappings (list of mappings for Bonita variable) that references others required objects such as: stringMatcherMapping and enumerationMatcherMapping, ...

Example of implementation classes for these objects are embedded into the bonita jar and defined into the environment.xml file.

4.3. Getting started with Bonita API

If you are already familiar with previous Bonita versions and you have already developped your own applications on top of Bonita, we want to minimize your effort when migrating to Bonita v4. Compatibility from Bonita v3 to v4 is our main concern. For this milestone Bonita has been packaged as

a java library and only plain java interfaces are provided (no session beans interface). Furthermore as there's no database persistence yet, all operations (including process deployment and execution) must be run in a single jvm.

4.3.1. XpdlImport interface

The XPDLImport facade class provides the following static method:

```
void    importDocument(final URL ressourceURL, final Environment environment)
```

4.3.2. User interface

Currently following methods are supported.

- To create an instance of the specified process

```
XpdlExecution instanciateProcess(final String processName)
```

- To obtain all user activities from specific instance

```
Collection<String> getToDoList(final String instanceName)
```

- To start an activity (when activity state is Ready). its state becomes Executing.

```
void startActivity(final String instanceName, final String nodeName)
```

- To terminate an activity (when activity is executing). Its state becomes Terminated

```
void terminateActivity(String instanceName, String nodeName)
```

4.3.3. Process interface

No method are yet implemented in this first Milestone.

4.4. Running the example

The Nova Bonita package contains one example of XPDL process under examples directory.

- BonitaSimpleProjectEver

The build.xml in the root directory contains the target to launch the example.

```
>ant examples
```

The java sample simply import the xpdl file of the process, then creates a workflow instance and leverage getTodoList, startActivity and terminateActivity methods from the User API