

Bonita Workflow

Process Console User's
Guide

BONITA WORKFLOW

Bonita Workflow

Process Console User's Guide

Bonita Workflow v3.1

Software

September 2007

Table of Contents

Chapter 1. Overview	1
1.1 Role of User	2
1.2 Role of Operator	2
1.3 Role of Administrator	2
Chapter 2. Process Console Description	3
2.1 Console loading	3
2.2 Console Frames Description	4
Chapter 3. Accessing and Creating Processes	7
3.1 How To Access the Workflow Process List	7
3.2 How To Create a New Instance of a Bonita Process	8
Chapter 4. Managing Manual Activities	9
4.1 Accessing the ToDo List	9
4.2 Starting an Activity	9
4.3 Viewing the List of "Running" Activities	10
4.4 Viewing the List of "Done" Activities	11
Chapter 5. Managing Process Models	13
5.1 How To Access Bonita Process Models List	13
5.2 Accessing Specific Model Information	14
5.3 How to Deploy/Undeploy a Bonita Process Model	15
5.4 How To Start a Process Model	16
5.5 How To Add/Suppress a User for a Process Model	17
5.6 Adding/Suppressing a User in a Role for a Process Model	18
5.7 Adding/Suppressing/Modifying a Role Mapper in a Process Model	19
Chapter 6. Managing Instances	21
6.1 How To Access the Instances List	21
6.2 Terminating or Deleting an Instance	22
6.3 Accessing Specific Instance Information	23

Chapter 7.	Managing Activities	25
7.1	Accessing the Activities List	25
7.2	Accessing Specific Activity Information.....	26
7.3	Displaying/Modifying/Deleting the Performer Assignment for an Activity.....	27
7.4	Starting/Canceling/Terminating an Activity	29
Chapter 8.	Monitoring.....	31
8.1	How To Modify the Setup for Bonita Monitoring.....	31
8.2	Accessing Logs	32
8.3	Accessing Traces for a Process Instance	33
8.4	Accessing the History of a Process Instance	34
Chapter 9.	Special “Super Operator” Recommendation.....	37
Chapter 10.	User Management.....	39
10.1	Bonita Workflow User Management Basic Configuration	39
10.2	Bonita Workflow Profiles.....	40
10.3	Changing the Basic Configuration	41
10.4	Initiating a New Datasource Security Realm for Bonita Workflow	43
10.4.1	Create the database	43
10.4.2	Adapt the JOnAS Datasource Configuration Files	43
10.5	Configuring the LDAP Directory for Bonita Workflow– LoginModule Feature	44
10.5.1	JOnAS Configuration	44
10.5.2	LDAP Configuration.....	45
10.5.3	Using the jonasAdmin Tool.....	47
10.5.4	LoginModule Feature	48
10.6	Selecting a New Security Realm for User Management	49
10.7	Accessing the description of a specific Realm	51
10.8	Accessing the Users List of a Specific Realm.....	52
10.9	Adding a User To a Specific Realm.....	53
10.10	Deleting a User From a Specific Realm.....	54
10.11	Accessing the List of Users Involved in a Specific Bonita Profile.....	55
10.12	Modifying Bonita Profiles or Passwords for a Specific User	56
Chapter 11.	Engine Databases	59
11.1	Changing the Basic Engine Datasource	59
11.2	Displaying the Engine Datasource Definition.....	59

Chapter 12.	Editing Custom Properties	63
Chapter 13.	Database Tuning	65
Chapter 14.	Database Migration Considerations	69

List of Figures

Figure 2-1. Console Login Screen	3
Figure 2-2. Process Console Frames	4
Figure 3-1. Bonita Processes that Have Been Deployed into the Workflow Engine	7
Figure 3-2. Process Properties Form	8
Figure 4-1. To Do List	9
Figure 4-2. Activity Properties Form	10
Figure 4-3. "Done" Activities Display.....	11
Figure 5-1. Deploy/Undeploy Window.....	14
Figure 5-2. Display of Available Process Models	14
Figure 5-3. Process Deploy / UnDeploy window	15
Figure 5-4. Starting a Process Model from the Process Model Display	16
Figure 5-5. Starting a Process from Process Display.....	16
Figure 5-6. Adding or Removing Process Users	17
Figure 5-7. Adding/Suppressing User Roles.....	18
Figure 5-8. Adding / Suppressing Users to/from Roles.....	18
Figure 5-9. Adding / Suppressing Role Mappers.....	19
Figure 5-10. Adding / Modifying Role Mapper.....	19
Figure 6-1. Displaying and Accessing Process Instances.....	21
Figure 6-2. Display of Information for a Specific Instance.....	23
Figure 7-1. Activities Display Window.....	25
Figure 7-2. Display of Information for a Specific Activity.....	26
Figure 7-3. Performer Assignment Display	27
Figure 7-4. Modify Performer Assignment Window.....	28
Figure 7-5. Starting/Terminating/Canceling Activities	29
Figure 8-1. Bonita Monitoring Setup Display	31
Figure 8-2. Bonita Log Records Display	32
Figure 8-3. Bonita Log Entries Detail.....	32
Figure 8-4. Bonita Trace Records Display	33
Figure 8-5. Bonita Trace Entries Details.....	33
Figure 8-6. Bonita History Display	34
Figure 8-7. History Instances for a Selected process model.....	34
Figure 8-8. History Instance Details	35
Figure 8-9. History Activity Details	35
Figure 10-1. Bonita Workflow User Management Basic Configuration.....	39
Figure 10-2. Bonita Workflow Profiles and Their Management Functions	40
Figure 10-3. Schema for Integrating A User Application into an Enterprise Information System.....	41
Figure 10-4. Datasource Realms Display in the Administration Console Core Frame.....	49
Figure 10-5. Confirmation Screen for Selecting a New Realm	50
Figure 10-6. Display of Available Realms	51

Figure 10-7. Detailed Information for an LDAP Realm.....	51
Figure 10-8. Display of Selected Realm	52
Figure 10-9. List of Users for a Selected Realm.....	52
Figure 10-10. New User Registration Form	53

Figure 10-11. Form for Removing Users from a Specific Realm.....	54
Figure 10-12. List of Available Profiles.....	55
Figure 10-13. List of Users for a Specific Profile	55
Figure 10-14. User Information Form.....	56
Figure 11-1. Bonita Datasource Configuration.....	60
Figure 12-1. Custom Properties Configuration Screen	63

Preface

This guide describes which facilities the Process Console provides to users via the User, Operator, and Administrator functions.

Chapter 1. Overview

This document describes the process console use, operations and administration for the Bonita Workflow. Although these activities may be performed by a single individual, the Process Console provides these facilities to users based on their role. This guide describes which facilities the process console provides for the User, the Operator, and Administrator roles.

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.
- Managing Manual Activities
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.
- Monitoring
Refer to Chapter 8.
- Special "Super Operator" Recommendation
Refer to Chapter 9.

For the Administrator

- User Management
Refer to Chapter 10.
- Engine Databases
Refer to Chapter 11.
- Editing Custom Properties
Refer to Chapter 12.
- Database Tuning
Refer to Chapter 13.
- Database Migration Considerations
Refer to Chapter 14.

1.1 Role of User

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

- Start Workflow Processes.
- Start / Stop / Cancel activities.
- Display terminated activities still visible.

1.2 Role of Operator

This guide provides the Operator with the information necessary to:

- Set User Preferences.
- Deploy, Undeploy, Start a Bonita process model.
- Terminate Bonita process model Instances.
- Access Bonita Process Model Instance information.
- Start, Terminate, or Cancel Bonita activity in a specific instance.
- Configure logs, trace, and history for Bonita Process Model Instances.
- Access logs, trace, and history files for a specific instance of a Bonita Process.

1.3 Role of Administrator

This guide provides the Administrator information necessary to:

- Modify the basic configuration for user management (LDAP or Datasource realm).
- For the Datasource Realm, Add or Remove users and Specify Bonita profiles for Bonita users.
- Access the Bonita Workflow engine Datasource configuration: JNDI name, port number.

For information about the different Bonita profiles and how to assign the administrator profile to a user, see the "Modifying Bonita Profiles or Passwords for a Specific User" section in this document.

Chapter 2.Process Console Description

2.1 Console loading

To access the Process Administration Console, connect to the following URL:

`http://Your_Host:Your_HttpPort/jiapAdmin`

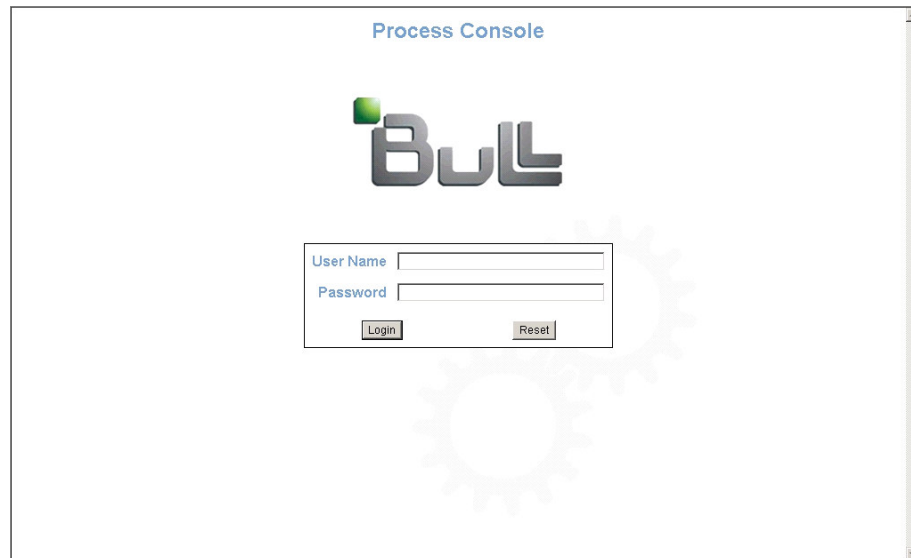


Figure 2-1. Console Login Screen

2.2 Console Frames Description

After logging in, the Administration Console is available in the main frame of a browser. It is divided into four parts (five if the footer frame is configured), each with a specific profile:

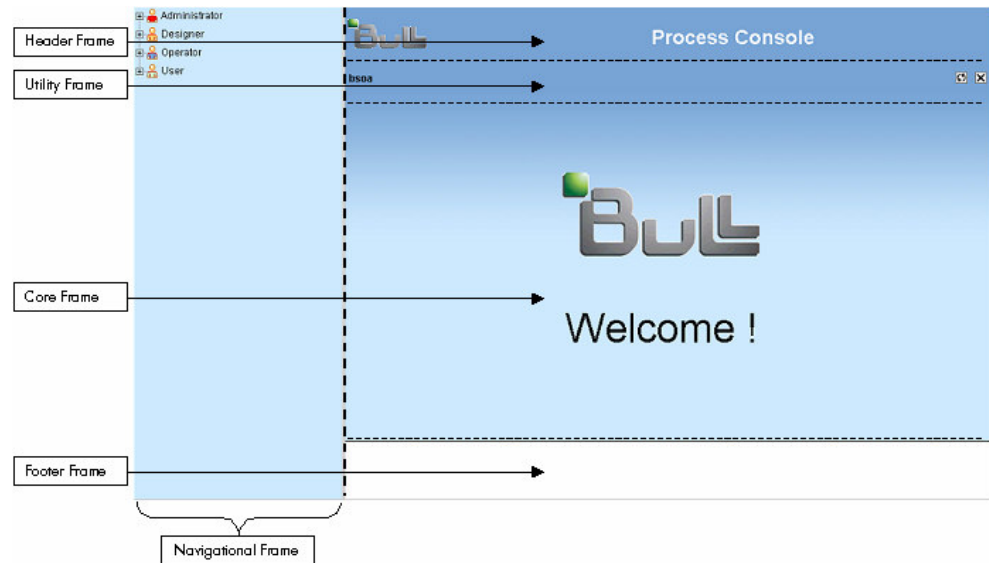


Figure 2-2. Process Console Frames

Navigational Tree

Use this frame to navigate between the different options the console offers, which are based on the user profile and the user-application context.

Click either on the / signs, or on the desired **labels** to expand/retract a branch.

For a terminal node, the **Core Frame** then presents the corresponding information.

Header Frame

By default, this frame displays the title and icon of the Process Console. The console administrator may customize the content of this frame by displaying the content of a configured URL. It also displays the path corresponding to the information shown in the Core Frame.

Utility Frame

This frame displays the name of the user logged in, a link to set user preferences, a button to refresh the header, core and footer frames, and a button to exit the console. It also displays the path corresponding to the information shown in the Core Frame.

Core Frame

This is the main frame of the console. A navigational path showing the tree structure is displayed along with available information. Different tabs may be accessed in this zone and all information will be displayed within it.

Footer Frame

By default this frame is not displayed. The console administrator may configure this frame to display the content of a configured URL (see **Customizing Header/Footer URLs** below).

Customizing Header/Footer URLs

To customize the header frame or display a customized footer frame, the properties **\$JONAS_BASE/conf/jiapadmin_custom.properties** file must be edited. This can be done either within the navigational tree by navigating to the **Edit Custom Frames** link of the **Administrator** tree, or by manually editing the above properties file.

The following is an example of a configured footer frame, while preserving the default header frame:

```
# Custom console fields

# bottom frame (footer) of the Admin Console.

footer=http://www.somewhere.com/acme.html

# top frame (header) of the Admin Console.

header=
```

It is the responsibility of the administrator to make sure the customized frame content fits within the dimensions of the frame.

Chapter 3. Accessing and Creating Processes

3.1 How To Access the Workflow Process List

Select the following path in the **Navigational Tree** (Left Panel): **User** → **Start** → **Project_name** as shown below in the example.

Details of the selected process are displayed in the **Core Frame** (Right Panel).

- **Name:** the name of the process
- **Version:** the version of the process
- **State:** current process life cycle state (i.e., READY, EXECUTING, TERMINATED, etc.)
- **Creator:** the login name of the designer importing the XPDL process definition into the workflow engine
- **Creation Date:** the date the process was imported
- **User Name:** the name of the user of the process
- **Role Name:** the role name of the process participant

Authenticated users (to the process console) can see all Bonita processes that are deployed by the workflow engine.

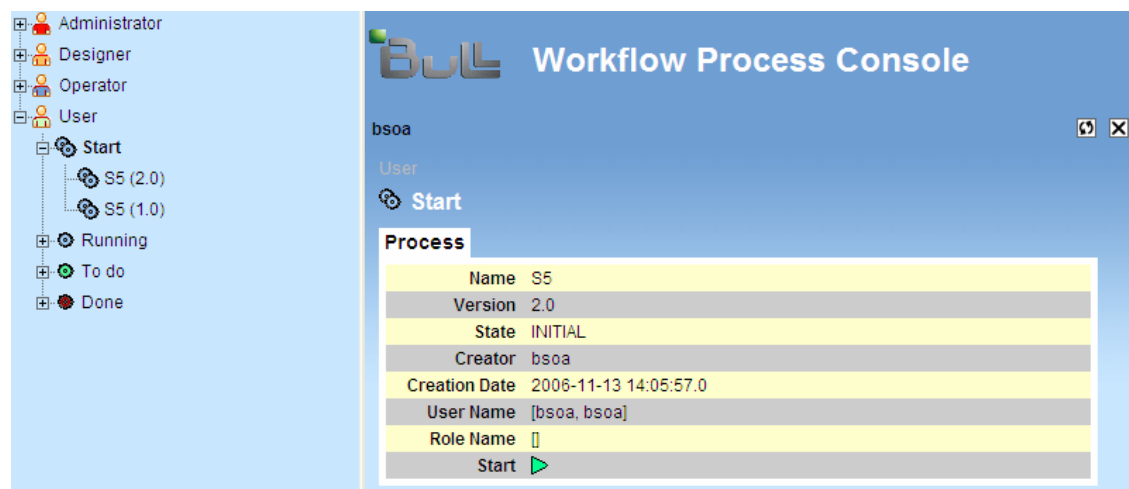



Figure 3-1. Bonita Processes that Have Been Deployed into the Workflow Engine

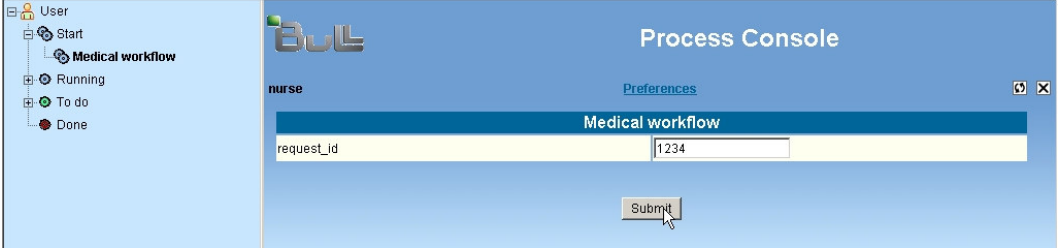
3.2 How To Create a New Instance of a Bonita Process

Select the following path in the **Navigational Tree** (Left Panel): **User** → **Start** → **Project_name** as shown below in the example.

To create a new instance of the process, click on the  (bottom “**Start**” line).

If the process has properties, a form is displayed within the core frame (right panel).

Fill in the form and click on the “**submit**” button as shown in the example below.



The screenshot displays the Bonita Process Console interface. On the left, the 'Navigational Tree' shows a hierarchy: 'User' (with a person icon) containing 'Start' (with a play icon) and 'Medical workflow' (with a document icon). Below 'Start' are 'Running', 'To do', and 'Done' (with a red dot icon). The main area is titled 'Process Console' and features a 'nurse' role indicator and a 'Preferences' link. A form titled 'Medical workflow' is shown with a single input field labeled 'request_id' containing the value '1234'. A 'Submit' button is located at the bottom of the form.

Figure 3-2. Process Properties Form

Chapter 4.Managing Manual Activities

The end-user process console manages manual activities under either the **READY**, **TERMINATED**, or **EXECUTING** state.

4.1 Accessing the ToDo List

Select the following path in the **Navigational Tree** (Left Panel): **User** → **To do**.

The list of activities waiting to perform is displayed in the **Core Frame**.

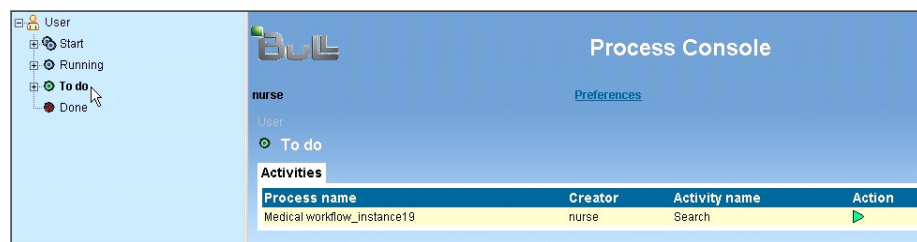


Figure 4-1. To Do List

4.2 Starting an Activity

Access the **ToDo** list of activities (see section 4.1).

The following parameters are displayed for each activity line:

- **Process name:** the name of the process and its instance number.
- **Creator:** the creator of the process instance.
- **Activity name:** the name of the activity within the process instance.

To start the activity, in the core frame:

- Click on the button at the end of the line of the activity you want to perform (as shown above on the example).

If this activity has properties to be set or read by the user, a form is displayed as shown in the example screen display below.

Fill in the form, then click on the “**Submit**” button as illustrated below.

The screenshot shows the 'Process Console' interface. On the left is a 'Navigational Tree' with a tree structure: 'User' (orange triangle icon) contains 'Start' (blue circle icon), 'Running' (blue circle icon), 'To do' (green circle icon), 'Medical workflow' (blue circle icon), 'Search' (green circle icon), and 'Done' (red circle icon). The main area is titled 'Process Console' and 'nurse'. It features a 'Preferences' link and a 'Medical workflow' section with a 'Search' input field. Below this are three input fields: 'request_id' with value '1234', 'init_date' with value '11/10/05', and 'end_date' with value '11/16/05'. A 'Submit' button is at the bottom right.

Medical workflow	
request_id	1234
init_date	11/10/05
end_date	11/16/05

Figure 4-2. Activity Properties Form

4.3 Viewing the List of “Running” Activities

Select the following path in the **Navigational Tree** (Left Panel): **User** → **Done** → **Project_name**.

The list of all current process instances and associated activities is displayed in the **Core Frame**.

4.4 Viewing the List of “Done” Activities

Select the following path in the **Navigational Tree** (Left Panel): **User** → **Running**.

The list of activities already performed is displayed in the **Core Frame**.

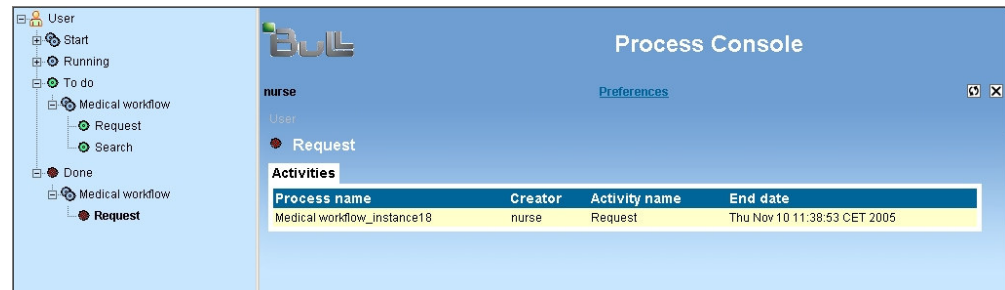


Figure 4-3. "Done" Activities Display

The following parameters are displayed for each activity line:

- **Process name:** the name of the process and its instance number.
- **Creator:** the creator of the process instance.
- **Activity name:** the name of the activity within the process instance.
- **End date:** the date on which the activity was “Done”.

Chapter 5.Managing Process Models

The user accessing the process console to see a given process model must have an **operator** profile. An Administrator level user of the process console sets the operator profile. See the *Process Console Administrator's Guide* to set operator profiles.

An operator can in his turn create other operator profiles within a created process. The first process operator is by default the designer who imported the process.

5.1 How To Access Bonita Process Models List

Select the following path in the **Navigational Tree** (Left Panel): **Operator → Process models**. The **Core Frame** (Right Panel) displays the available process models.

For each process model, its type (Bonita), its state, and its creator is displayed. Depending on its state, various actions are available.

- **Start**

When an XPDL process definition file is imported in the Bonita engine, the state of the process is automatically deployed, meaning the users can start the process model.

- **UnDeploy**

The UnDeploy action prevents end users from starting new instances of the process. The operator can make modifications on the process (for instance: adding/modifying role mappers).

When this model is returned to "Deployed", any changes made on the process are implemented for new process instances. Instances created before the "Undeploy" actions maintain the initial definition of the process model.

Instead of making changes on the process model, "Undeploy" can be used to import a new version of the process (in this case, all process instances must be terminated or canceled before the process model is deleted)

- **Deploy**

The Deploy action can logically follow the "Undeploy" action. It returns the capability of starting a process to end users.

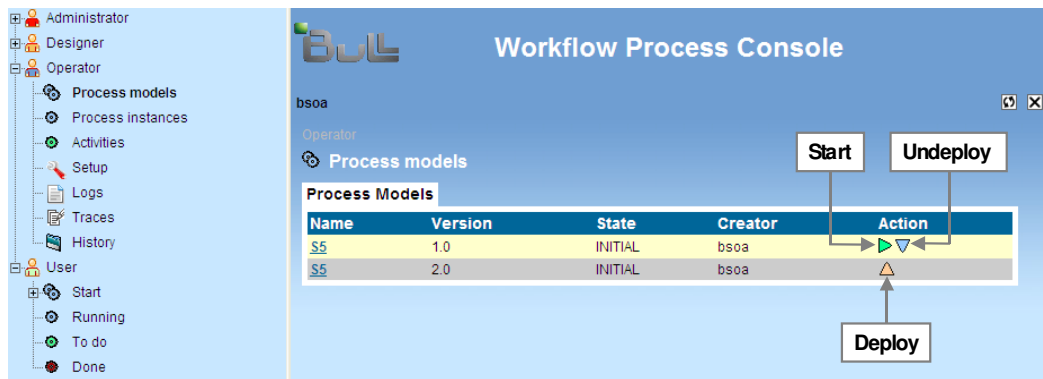


Figure 5-1. Deploy/Undeploy Window

5.2 Accessing Specific Model Information

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

To select a process and see its specific information, select it in the list by clicking on its name in the **Core Frame**.

The following information is displayed.

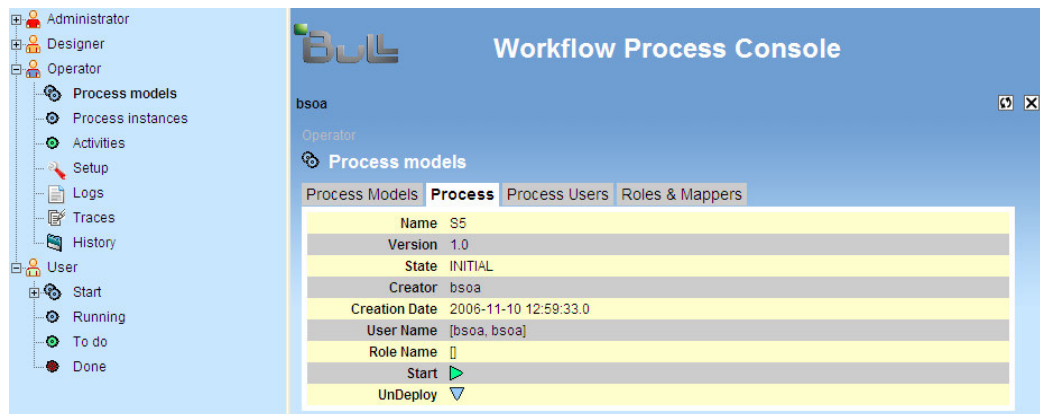


Figure 5-2. Display of Available Process Models

- **Process Name:** the name of the process model.
- **Version:** the version of the process model.
- **State:** the state of the process model (INITIAL, DEPLOYED, TODEPLOY).
- **Creator:** the login name of the designer importing the XPDL process definition into the workflow engine.
- **Creation Date:** the date and time the process model was created.
- **User Name:** the Human Participants Name (user login) list defined in the process.
- **Role Name:** the Role Participants Name defined in the process.
- **Deploy or (UnDeploy and Start):** available actions as explained in Section 5.1.

5.3 How to Deploy/Undeploy a Bonita Process Model

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

The first way to deploy/undeploy a process model is to click on the chosen action in the right column of the **Core Frame** (named “**Action**”): Deploy or UnDeploy, depending the current state of the process.

The second way is selecting the model in the Model List by clicking on its name, and clicking on the corresponding button (Deploy, UnDeploy) shown in the **Core Frame**.

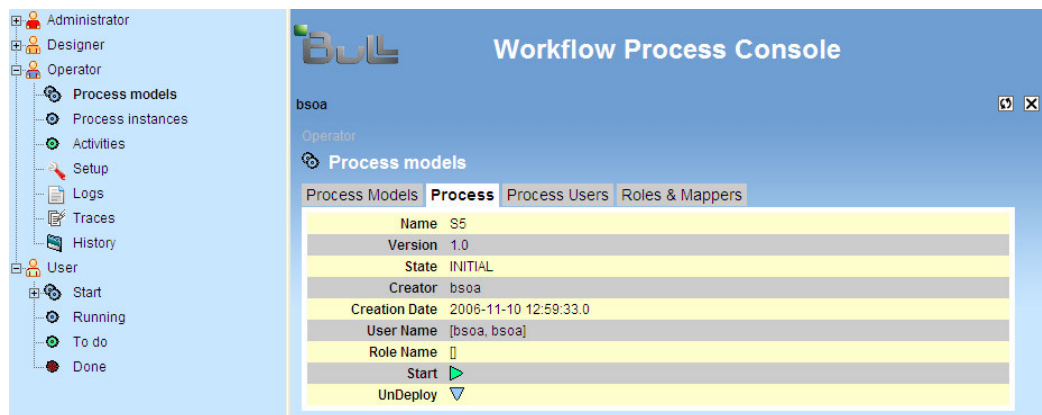


Figure 5-3. Process Deploy / UnDeploy window

5.4 How To Start a Process Model

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

The first way to instantiate a process model is to click on the **"Start"** (▶) action on the right column of the **Core Frame** (named "Action"). This action is available for deployed process models only.

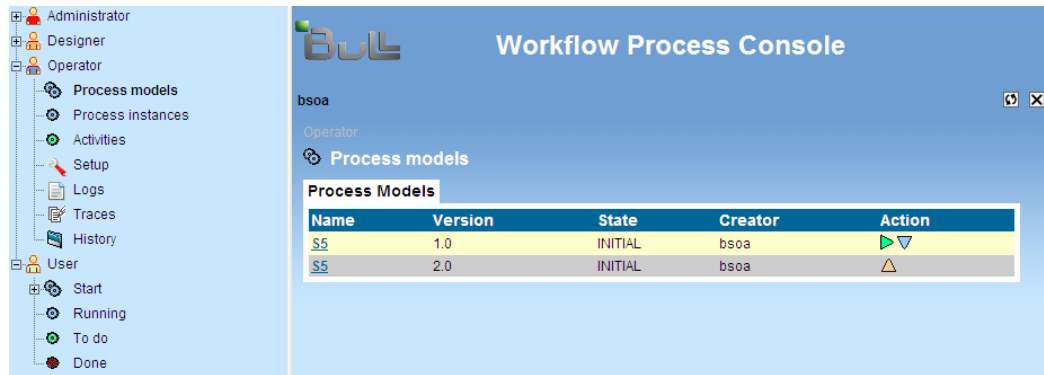


Figure 5-4. Starting a Process Model from the Process Model Display

The second way is selecting the model in the Process Models List by clicking on its name, and clicking on the corresponding ▶ button (**"Start"**) shown in the back of the **Core Frame**.

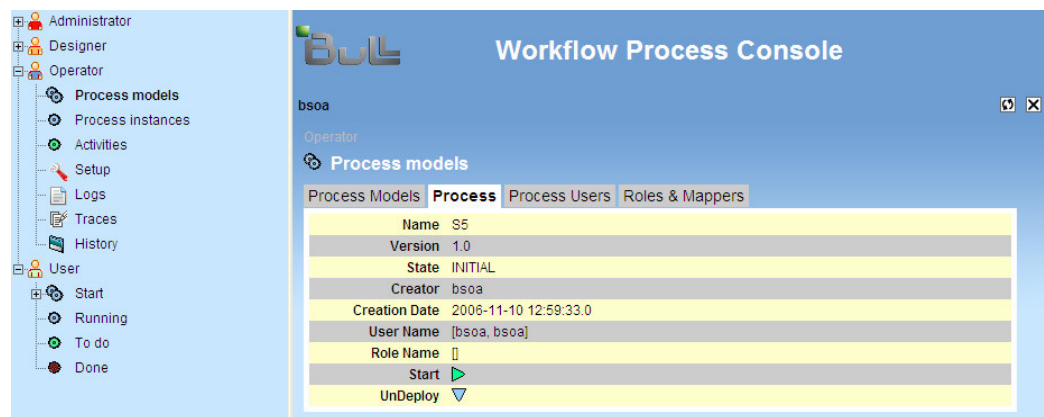


Figure 5-5. Starting a Process from Process Display

If the process has properties, a form is displayed in the core frame (right panel).

Fill in the form and then click on **"Submit"**.

5.5 How To Add/Suppress a User for a Process Model

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

Select a process by clicking on its name in the **Core Frame** and select the "Process Users" tab by clicking on it.

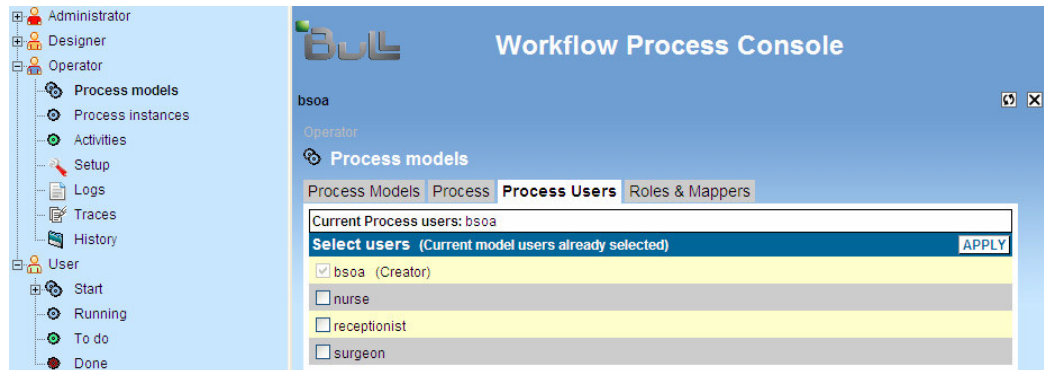


Figure 5-6. Adding or Removing Process Users

To add a user to the process participants: Select the checkbox in front of the user being added, then click on the "Apply" button.

To suppress the user from the process participants: Deselect the checkbox in front of the user being suppressed, then click on the "Apply" button.

Adding and suppressing users in the process is a required feature for the following Section 3.1.6 (add/suppress a user in a role).

5.6 Adding/Suppressing a User in a Role for a Process Model

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

First select a process by clicking on its name in the **Core Frame**. Then select the **“Role & Mappers”** tab by clicking on it.

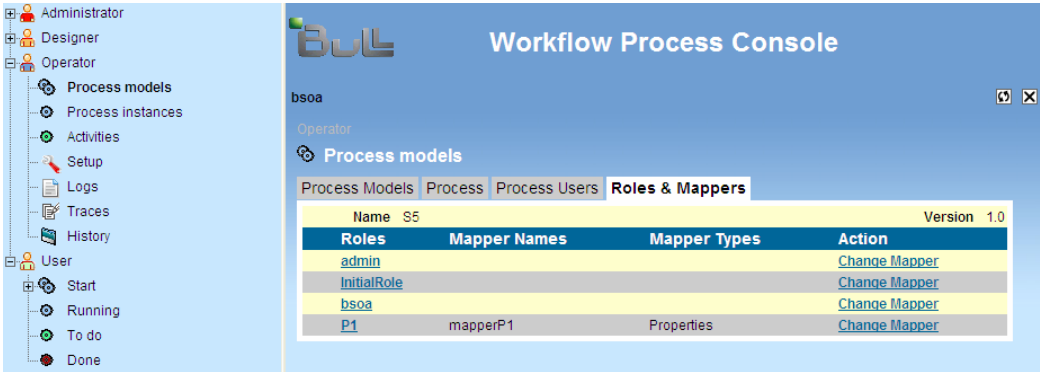


Figure 5-7. Adding/Suppressing User Roles

First select the role that is to be modified by clicking on its name in the **Core Frame** (Right Panel). The detail of the users involved in this role is then displayed.

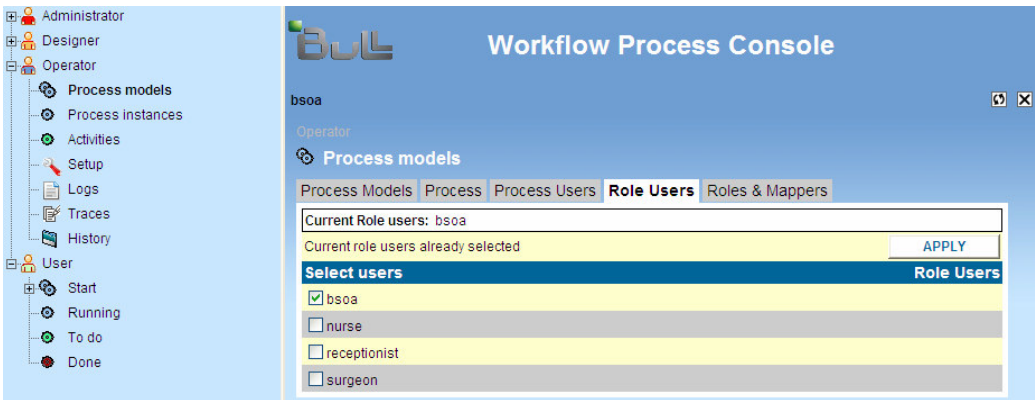


Figure 5-8. Adding / Suppressing Users to/from Roles

To add a user to the role: Select the checkbox in front of the user to be added to the role, then click on the **“Apply”** button.

To suppress a user from the role: Deselect the checkbox in front of the user to be suppressed from the role, then click on the **“Apply”** button.

5.7 Adding/Suppressing/Modifying a Role Mapper in a Process Model

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process models**. The **Core Frame** (Right Panel) displays the available process models.

First select a process by clicking on its name in the **Core Frame**. Then select the **“Role & Mappers”** tab by clicking on it.

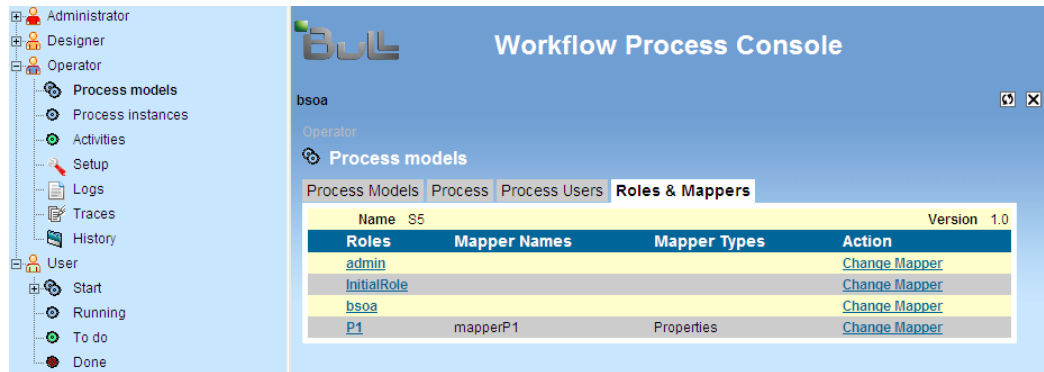


Figure 5-9. Adding / Suppressing Role Mappers

Select the **“Change Mapper”** action on the chosen Role by clicking on the Action in the corresponding line. The following screen is displayed in the core frame.

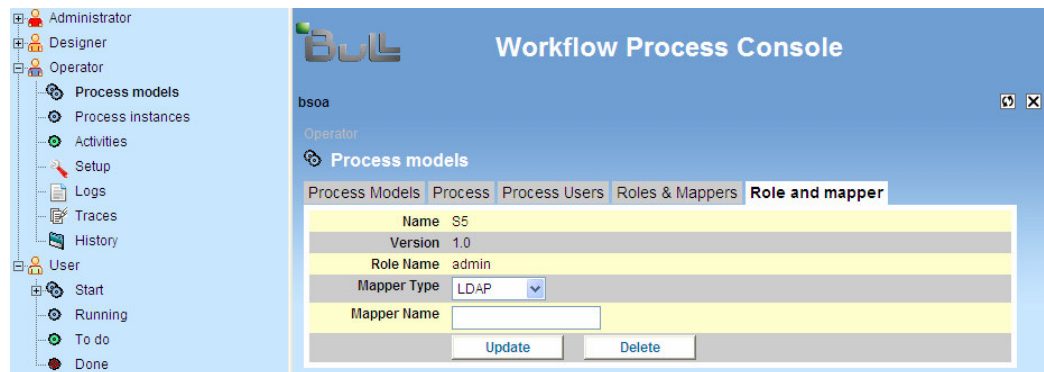


Figure 5-10. Adding / Modifying Role Mapper

To add or modify a Role Mapper:

- Click on the "Mapper Type" drop down menu.
- Three types of mappers are available: LDAP, Properties, Custom. Select the Mapper Type.
- Fill in the Mapper Name.
- Click on the **"Update"** button.

To remove a Role Mapper: Click on the "Delete" button.

Chapter 6.Managing Instances

The requirements that must be met by a user accessing the process console in order to see process instances for a given process model are the same as those described in Chapter 5.

6.1 How To Access the Instances List

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process instances**. The **Core Frame** (Right Panel) displays the available process instances.

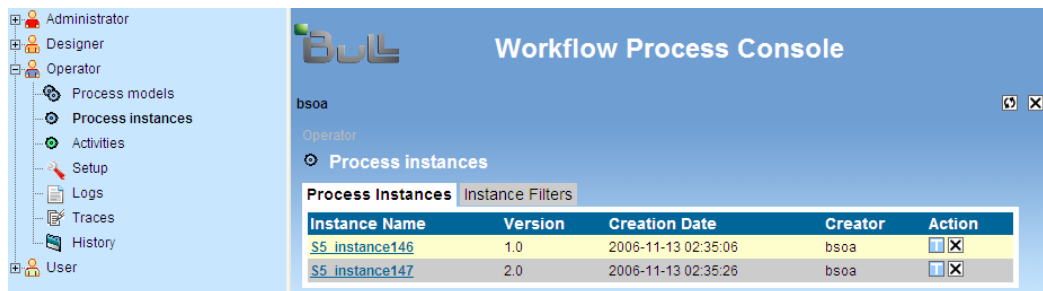


Figure 6-1. Displaying and Accessing Process Instances

The following information is available:

- **Instance Name:** the name of the process instance.
- **Creator:** the login name of the end user creating the process instance.
- **Version:** the version of the process instance.
- **Creation Date:** the date and time the process instance was created.
- **Action:** **Terminate action** (Actions requiring specific conditions to be successful are explained in Chapter 6.2). **Delete action:** this action removes the Instance name from the process instances.

6.2 Terminating or Deleting an Instance

Select the following path in the **Navigational Tree** (Left Panel): **Operator → Process instances**. The **Core Frame** (Right Panel) displays the available process instances.

Click on the **"T"** Button in front of the chosen instance to terminate it.

Terminating a process instance is only possible if there is no activity in the **"READY"** or **"EXECUTING"** state.

If the **"Terminate"** action is successful, the **Core Frame** (Right Panel) displays the available process instances. Otherwise an error message is returned i.e.: *"Project cannot be terminated. Some nodes are still active"*.



Note:

An activity may be in the **EXECUTING** state if an error occurred during java hook execution. If the operator decides to cancel the activity, then the activity state becomes **DEAD** and then it is possible for the operator to terminate the instance. A History of the instance is added to the history log.

Click on the **"X"** Button in front of the chosen instance to remove it from the **"Process Instances"** view.

6.3 Accessing Specific Instance Information

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Process instances**. The **Core Frame** (Right Panel) displays the available process instances.

Then select an instance by clicking on its name in the **Core Frame**.

The following screen is displayed.

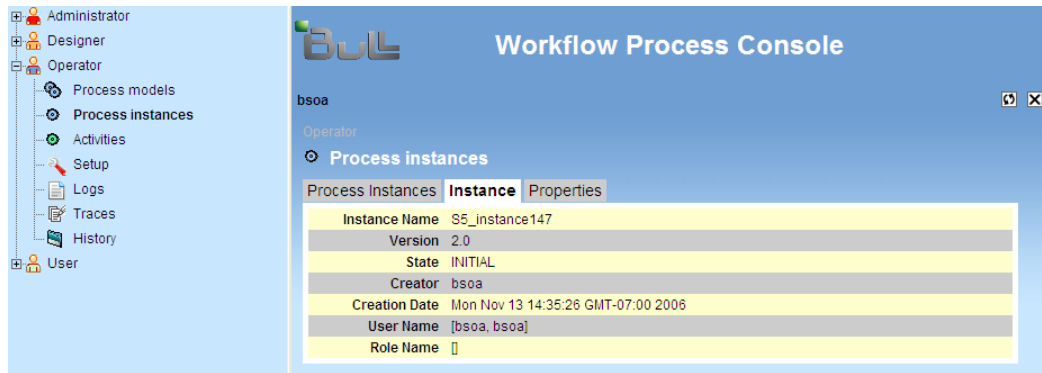


Figure 6-2. Display of Information for a Specific Instance

The following information is available:

- **Instance Name:** the name of the process instance.
- **Version:** the version of the process instance
- **State:** the state of the process instance (READY).
- **Creator:** the login name of the designer who imported the XPD L process definition into the workflow engine.
- **Creation Date:** the date and time the process instance was created.
- **User Name:** the Human Participants Name (user login) list defined in the process.
- **Role Name:** the Role Participants Name defined in the process.

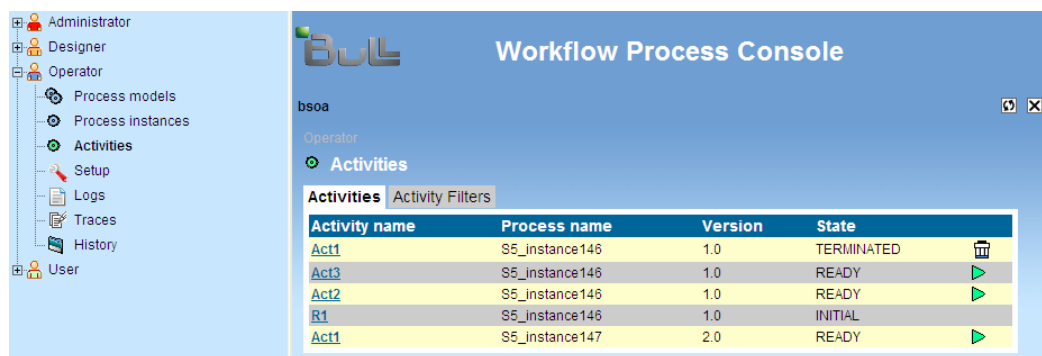
Chapter 7.Managing Activities

User requirements for accessing the process console to view activities from all “not terminated” instances are the same as those described in Chapter 5.

7.1 Accessing the Activities List

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Activities**. The **Core Frame** (Right Panel) displays the available activities.

Activities are grouped by process instance and in order of the sequenced flow of the process inside each group.



The screenshot shows the Workflow Process Console interface. On the left is the Navigational Tree with the following structure: Administrator, Designer, Operator (selected), Process models, Process instances, Activities (selected), Setup, Logs, Traces, History, and User. The main panel on the right is titled 'Workflow Process Console' and shows the 'Activities' tab. It displays a table of activities with columns: Activity name, Process name, Version, State, and an Action icon. The table contains five rows of activity data.

Activity name	Process name	Version	State	Action
Act1	S5_instance146	1.0	TERMINATED	🗑️
Act3	S5_instance146	1.0	READY	▶️
Act2	S5_instance146	1.0	READY	▶️
R1	S5_instance146	1.0	INITIAL	▶️
Act1	S5_instance147	2.0	READY	▶️

Figure 7-1. Activities Display Window

Following information is available:

- **Activity name:** the name of activity.
- **Process name:** the name of the process model.
- **Version:** the version of the process model.
- **State:** state of the activity (INITIAL, READY, TERMINATED, EXECUTING, DEAD).
- **Action:** Start, Cancel, Terminate.

7.2 Accessing Specific Activity Information

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Activities**. The **Core Frame** (Right Panel) displays the available activities.

Then select an activity by clicking on its name in the **Core Frame**. The “**Activity Properties**” tab is then displayed.

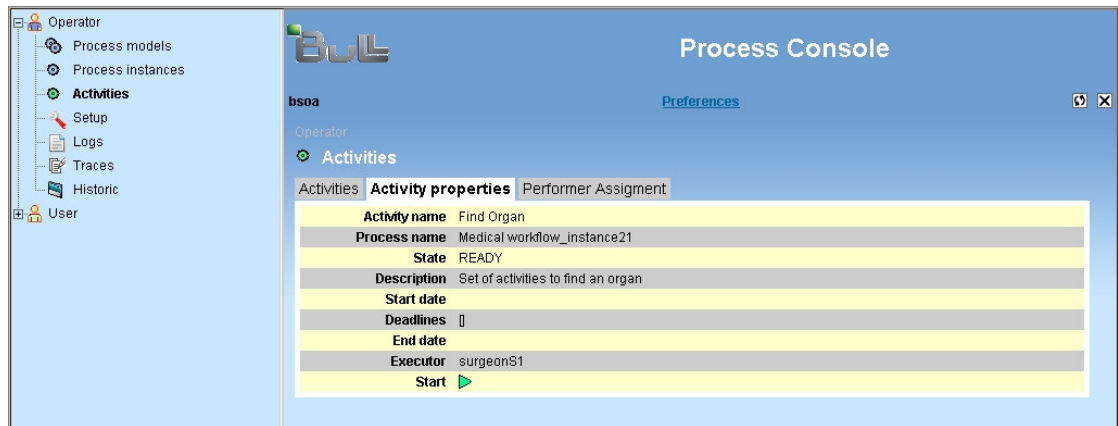


Figure 7-2. Display of Information for a Specific Activity

The following information is available:

- **Activity name:** the name of the activity.
- **Process name:** the name of the process model.
- **State:** the state of the activity (READY, EXECUTING, TERMINATED...)
- **Description:** the description of the activity set in the XPD activity definition.
- **Start date:** the date the activity started.
- **Deadlines:** the list of deadline names set in the XPD activity definition.
- **End date:** the date the activity has been terminated.
- **Executor:** the name of the role the activity is assigned.
- **Start** (State= READY) or **Remove** (State=TERMINATED).



Note:

If an activity is in the “INITIAL” or “DEAD” state, this **Start** line is not displayed.

7.3 Displaying/Modifying/Deleting the Performer Assignment for an Activity

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Activities**. The **Core Frame** (Right Panel) displays the available activities.

Then select an activity by clicking on its name in the **Core Frame**.

To **Display** the Performer Assignment, click on the "**Performer Assignment**" tab.

If the Performer Assignment has been set, the following information is displayed:

- **Activity Name:** the name of the activity.
- **Current Performer:** the name of the current performer. If empty then no performer is currently assigned.
- **Select Performer:** select performer name. If blank selection is made then no current performer is assigned.

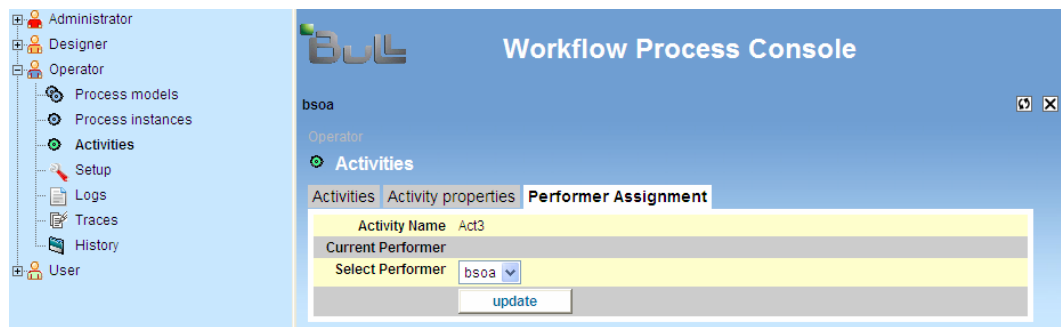


Figure 7-3. Performer Assignment Display

To modify the Performer Assignment, click on the **"Modify"** link as shown in Figure 3-15. The following window displays for modifying the Performer assignment information.

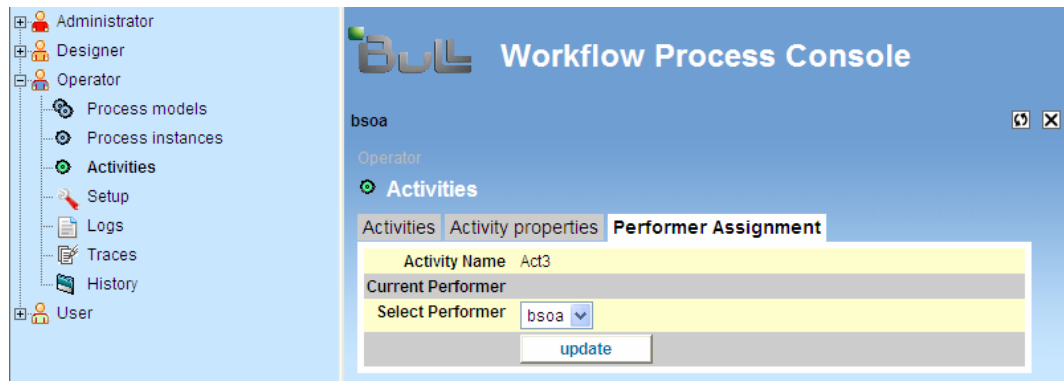


Figure 7-4. Modify Performer Assignment Window

- Fill in the Performer Assignment form (with information as described above).
- Click on **"update"** to validate.

To **delete** the Performer Assignment, click on the **"modify"** link and then click on **"delete"**.

7.4 Starting/Canceling/Terminating an Activity

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Activities**. The **Core Frame** (Right Panel) displays the available activities.

Then select a command (Start, Cancel, Terminate), by clicking on the corresponding button at the end of the chosen activity line.

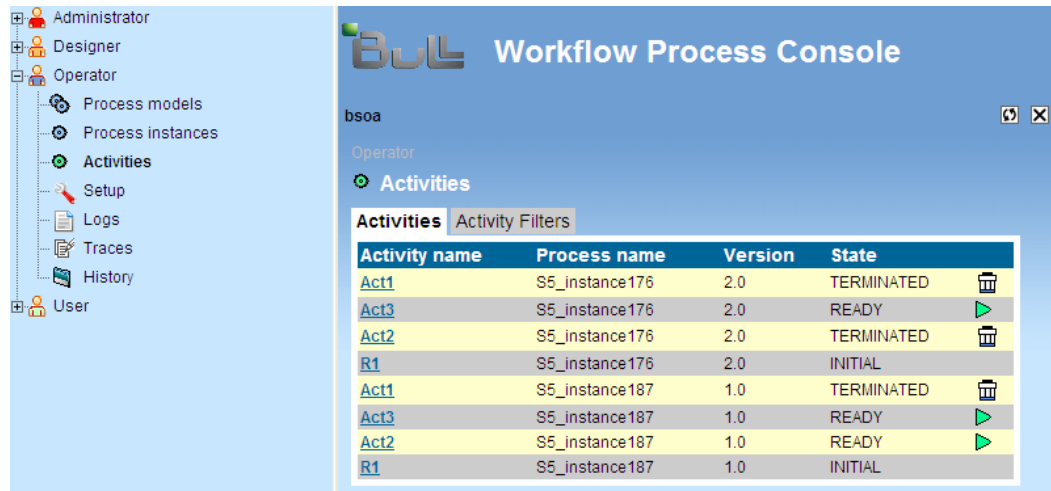


Figure 7-5. Starting/Terminating/Canceling Activities

When clicking on **"Start"**, a form is displayed if this activity has properties to be set or read. Fill in the form and click on the **"Submit"** button to validate it. The activity goes on and terminates automatically if no error occurs during execution (especially if hooks have been set).

In the case where an activity execution fails, its state becomes **"EXECUTING"**. **"Terminate"** and **"Cancel"** buttons appear on the activity line of the activity list.

When clicking on **"Cancel"**, the activity execution is stopped and its state changes to **"DEAD"**.

When clicking on **"Terminate"**, activity execution is launched again and terminates automatically if no error occurs during execution.

Chapter 8. Monitoring

The Setup tree manages the configuration of four Bonita features:

- Java Messaging Service ("**JMS**") used by Bonita workflow.
- Bonita technical traces ("**Trace**").
- Bonita log ("**Log**").
- The creation of history entries for instances whenever they are terminated ("**History**").

The default setup configuration is:

- JMS: Disable
- Trace: Info level
- Log: Info level
- History: Archive

Available setup configuration is:

- JMS: Disable, Enable
- Trace: Info, Debug, Error level
- Log: Info, Debug, Error level
- History: Archive, Purge

8.1 How To Modify the Setup for Bonita Monitoring

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Setup**.

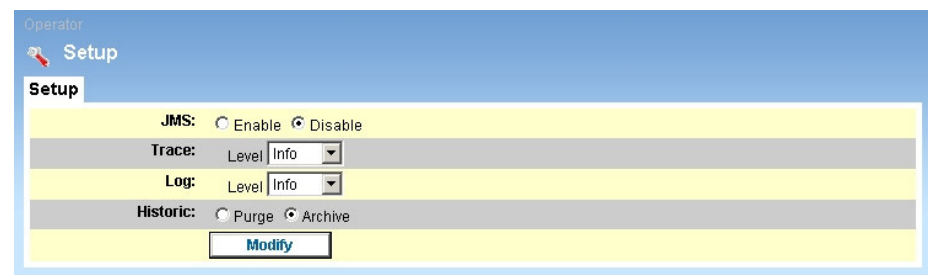


Figure 8-1. Bonita Monitoring Setup Display

- Choose the configuration setup for Bonita features: JMS, Trace, Log, History.
- Click on the "**Modify**" button.
- Confirm this modification by clicking on the "**Confirm**" button.

8.2 Accessing Logs

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Logs**.

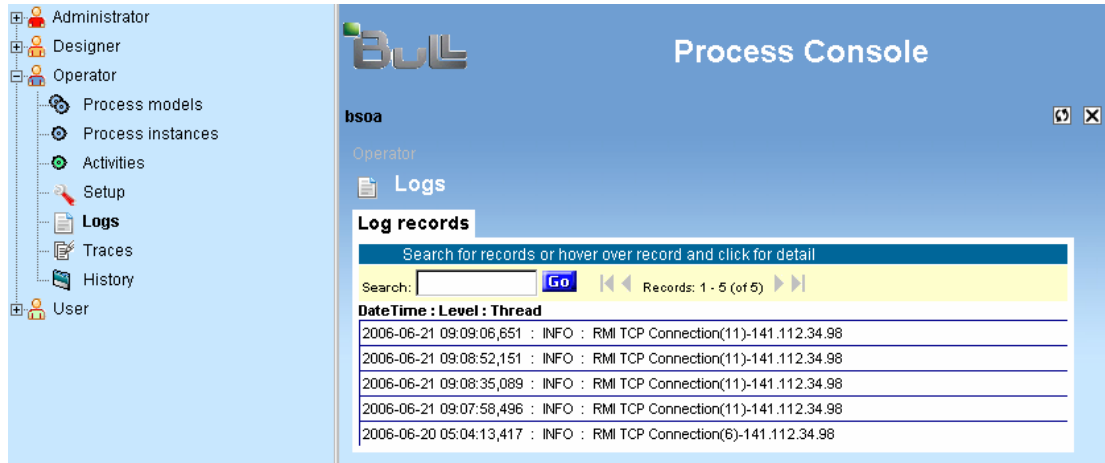


Figure 8-2. Bonita Log Records Display

Some utilities are available within the log screen:

- **Search:** Enter an expression into the text area and click “Go”.
- Navigate entry pages (entries are displayed 10 by 10).
- Display the first and last entry pages.

Click a line of the log table to get details of this log line as shown in the following figure.

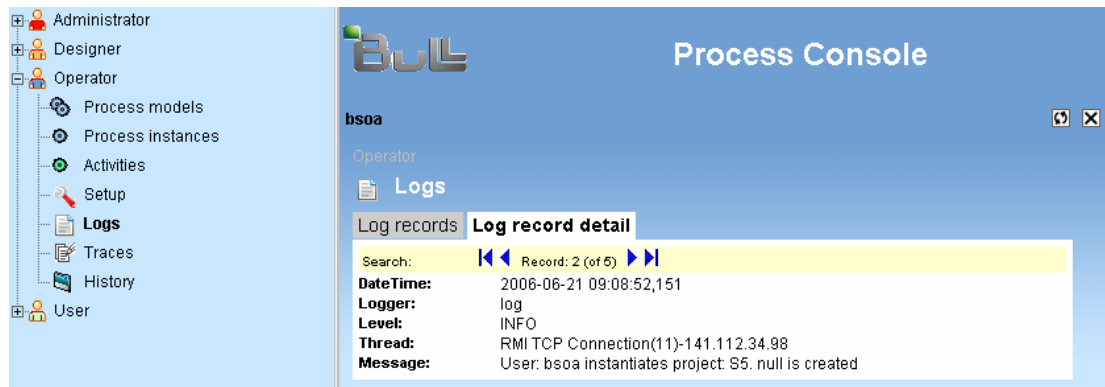


Figure 8-3. Bonita Log Entries Detail

8.3 Accessing Traces for a Process Instance

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **Traces**.

The core frame displays the trace records of methods used when a process is launched.

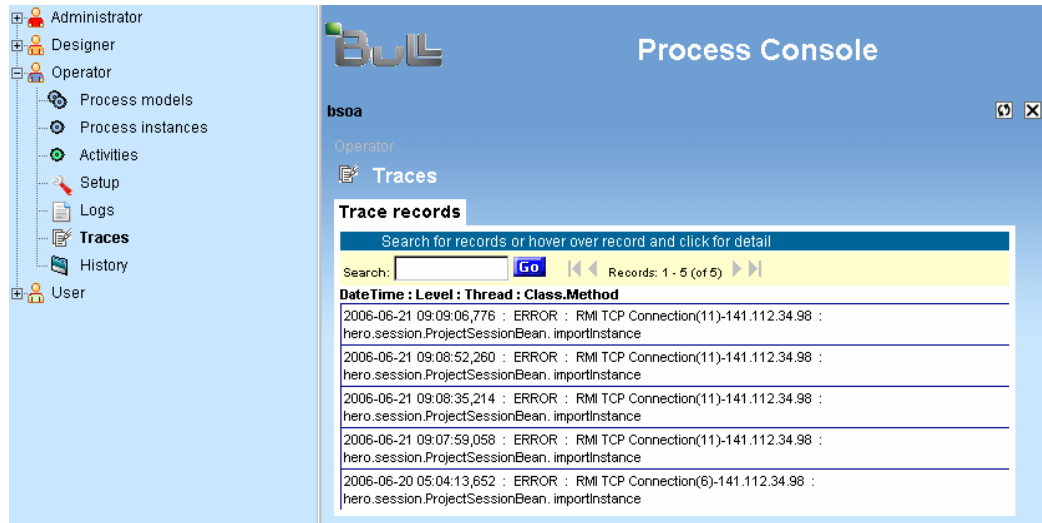


Figure 8-4. Bonita Trace Records Display

Some utilities are available within the Traces screen:

- **Search:** Enter an expression into the text area and click **"Go"**.
- Navigate entry pages (entries are displayed 10 by 10).
- Display the first and last entry pages.

Click a line of the traces table to display details of this trace line as shown in the following figure.

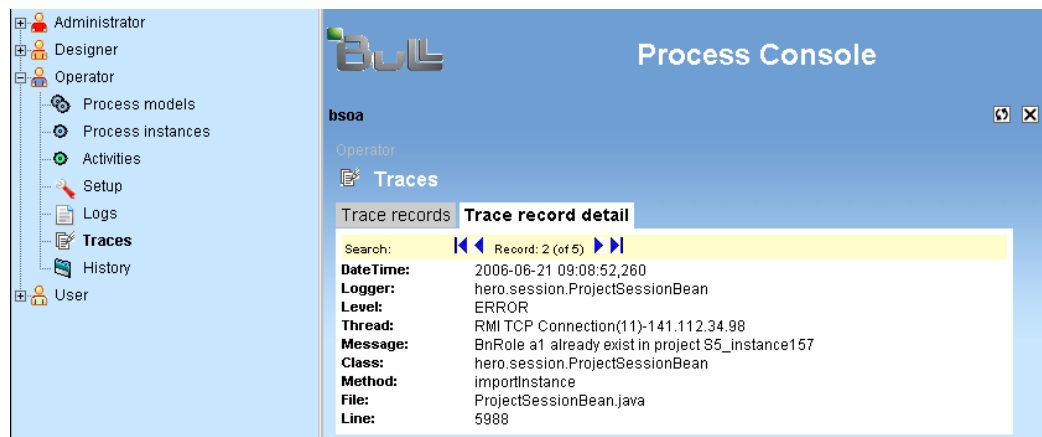


Figure 8-5. Bonita Trace Entries Details

8.4 Accessing the History of a Process Instance

Select the following path in the **Navigational Tree** (Left Panel): **Operator** → **History**.

The core frame displays the list of process models (the following figure displays only one).

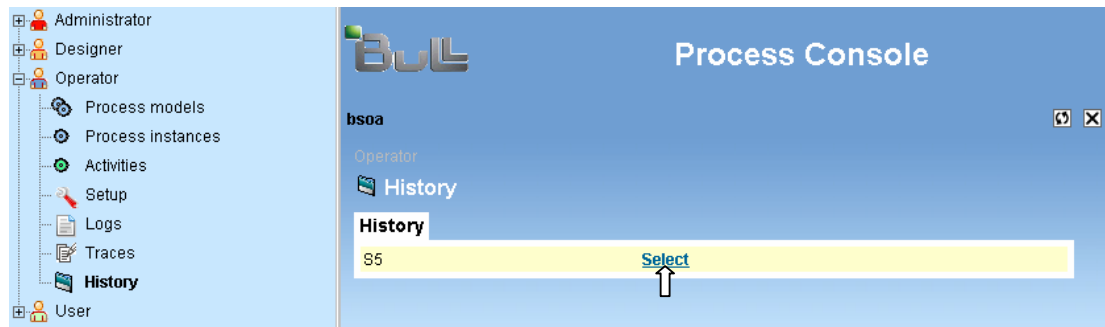


Figure 8-6. Bonita History Display

- Click on the “**Select**” link on the line with the chosen process model.
- The core frame then displays the list of instances for the chosen process model.

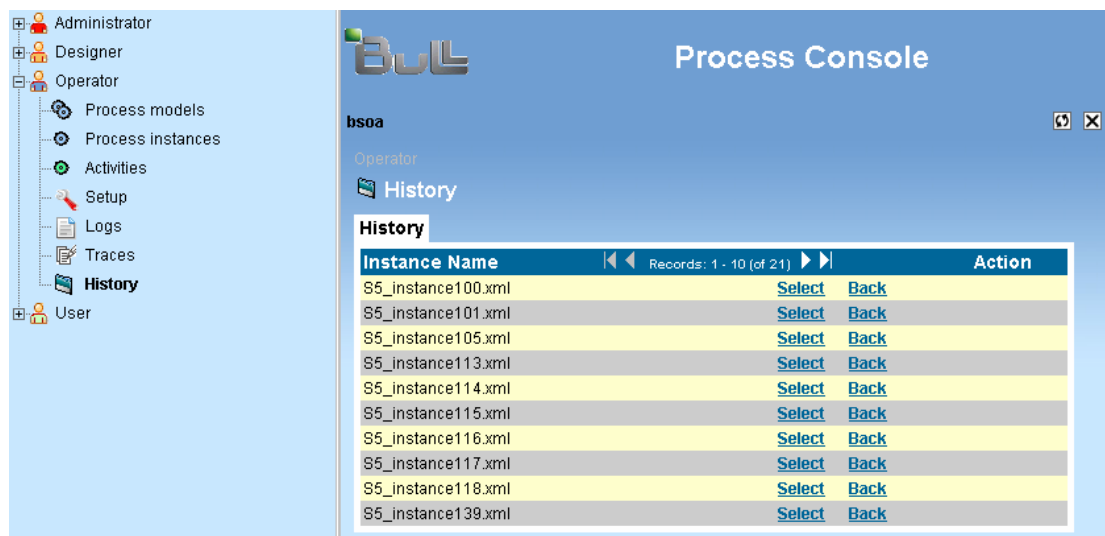


Figure 8-7. History Instances for a Selected process model

- Click on the **"Select"** link on the line with the chosen process instance. Click on the **"Back"** link to return to the list of process models.
- The history of the process instance is displayed inside the core frame.

Process Console

History

Instance Name **Action**

Instance Name: S5_instance105
Creation Date: 2006-06-07 16:33:59.0
End Date: Mon Jun 12 13:58:38 MST 2006
Initiator: a1

[Back](#)

Activities	Executor	Start Date
Act1	a1	2006-06-12 13:34:45.0
Act3	a1	Mon Jun 12 13:56:15 MST 2006
Act2	a1	2006-06-12 13:35:36.0
R1	a1	Mon Jun 12 13:56:16 MST 2006
Act4	a1	Mon Jun 12 13:58:38 MST 2006

Figure 8-8. History Instance Details

- Click a cell containing an activity to view its details.

Process Console

History

Activity Name **Action**

Activity Name: Act2
State: TERMINATED
Role: P1
Executor: a1
Type: AND_JOIN_NODE
Anticipable: false
Description:
Start Date: 2006-06-12 13:35:36.0
End Date: 2006-06-12 13:35:36.0

[Back](#)

Figure 8-9. History Activity Details

Chapter 9.Special “Super Operator” Recommendation



Important:

Add a “super” Operator for each process model of the workflow engine.

When the Designer (user granted the JIAPDESIGNER profile for the process console) imports the process model into the workflow engine, the Designer is automatically granted the “admin” role for the imported process into the engine database.

This “admin” role is mandatory to allow an Operator (user granted the JIAPOPERATOR profile for the process console) to be able to access the process model.

It is strongly recommended that a “super” operator be added for each imported process model. If the initial Designer is removed from the authentication user base, then the Super Operator can continue to operate on the process. Otherwise, it would be necessary to add the initial Designer back into the user authentication base to be able to operate on the process.

To add a “super” Operator for each process model of the workflow engine, perform the following two steps for each imported process model.

1. Give the “admin” role of the workflow engine to a chosen user for the process model.
For the process model, execute the “Add User” operations described in Section 5.5 of this guide. The user chosen to be the super operator should have been previously deployed into the user's base.
One option is to deploy the generic user “bsoa” into the user base.
Add this user in the role of “admin” for the process model. Refer to Section 5.6 of this guide.
2. Give the Operator profile of the process console to this chosen user.
An Administrator (user granted to JIAPADMINISTRATOR profile for the process console) should add the JIAPADMINISTRATOR profile to the chosen user. (See Section 3.12, Modifying Bonita Profiles or Passwords for a Specific User.)

Chapter 10. User Management

10.1 Bonita Workflow User Management Basic Configuration

After Bonita Workflow installation, specific information for user management is stored in the default security Datasource realm, as shown below. This Datasource points to an HSQL database, containing the Administration console and the Bonita Workflow engine data.

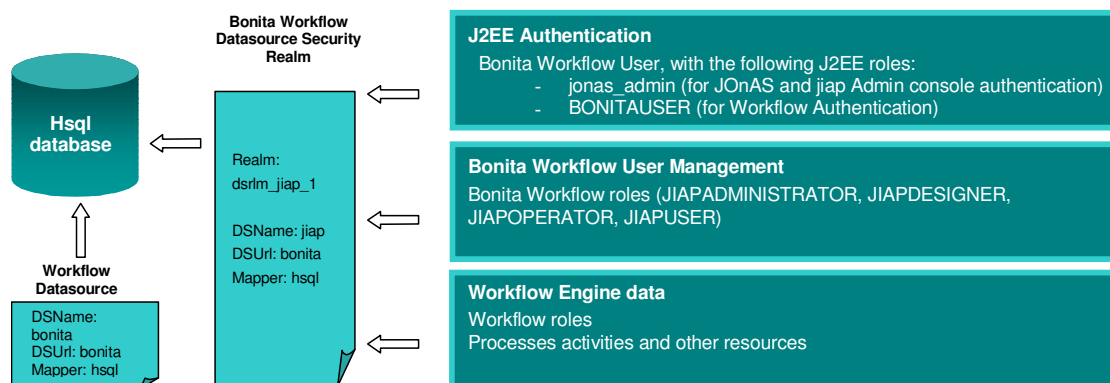


Figure 10-1. Bonita Workflow User Management Basic Configuration

The Bonita Workflow installation process:

- Creates and initiates the Datasources (creates both `jiapadmin.properties` and `bonita.properties` files).
- Adds the associated Datasource security realm (updates `server.xml` and `jonas-realm.xml` files).

This basic configuration can be changed according to user preferences: for example, using the enterprise LDAP directory or moving to another security Datasource realm.

10.2 Bonita Workflow Profiles

There are four different profiles that cover the main functions of the Workflow management can be granted to Bonita Workflow users.

The following figure displays the scope of each one.

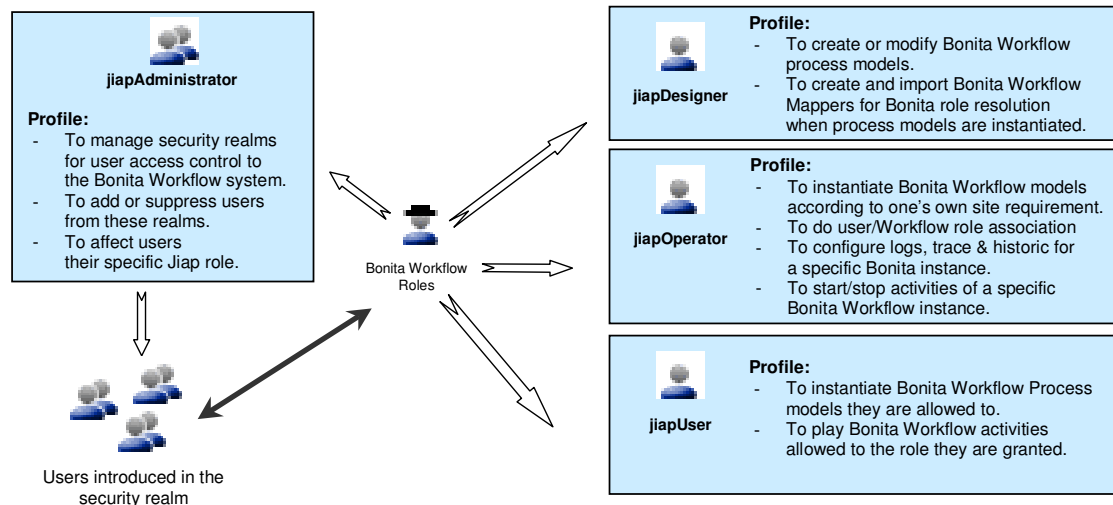


Figure 10-2. Bonita Workflow Profiles and Their Management Functions

10.3 Changing the Basic Configuration

User Management can move to the following schema to make a user application fully integrated into the enterprise Information System. Bonita Workflow takes advantage of User Management defined at an upper level to fit the needs of the user's Workflow application.

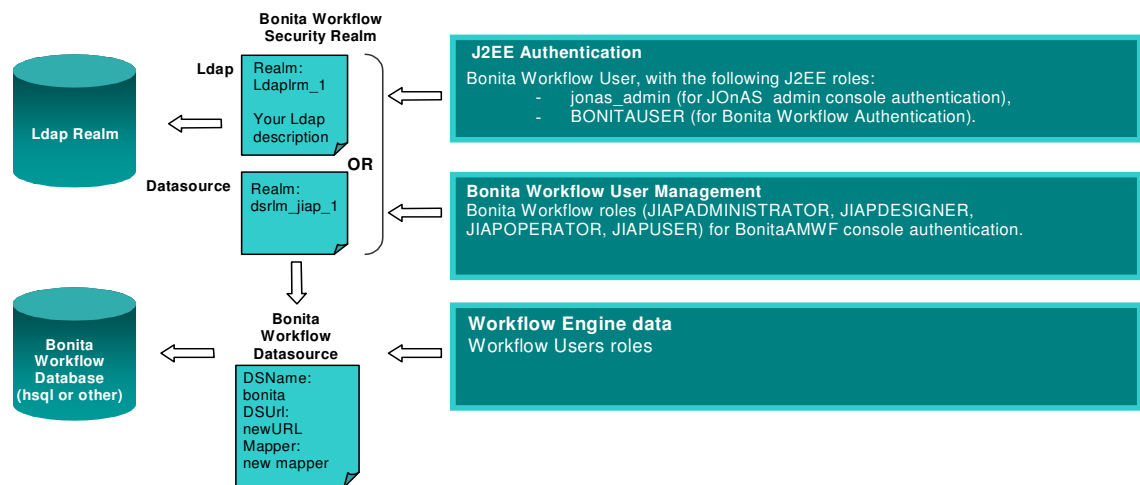


Figure 10-3. Schema for Integrating A User Application into an Enterprise Information System

Bonita Workflow uses the security realm defined in the global context for JOnAS. To change the basic configuration:

Using another Datasource Security Realm:

- Create the new database and adapt a Datasource description file (jiapadmin.properties) with the new URL.
- Initiate this Datasource appropriately for use with Bonita Workflow (See the "Initiating a New Datasource Security Realm for Bonita Workflow" section below).
- Open a DOS command line or a Linux terminal console and enter the path of the Bonita Workflow installation directory, and enter either the "configure.bat" command for Windows, or the "configure.sh" command for Linux.

Using an LDAP Security Realm

- Create an LDAP Directory, if it does not exist already.
- Configure this directory appropriately for use with Bonita Workflow (See the "Configuring the LDAP Directory for Bonita Workflow– LoginModule Feature" section below).
- Open a DOS command line or a Linux terminal console and enter the path of the Bonita Workflow installation directory and enter either the "configure.bat" command for windows or the "configure.sh" command for Linux.

At this point, the JOnAS Application server uses the user Datasource or LDAP Security Realm. The last step is informing Bonita Workflow to use this for its own user management.

To do this:

- Use the Process console to change the User Management realm (See the "Selecting a New Security Realm for User Management" section below)
- Stop and restart the JOnAS Server.



Note:

Datasource or LDAP Realm configuration parameters are accessed and modified by using the **JOnAS** Admin Console http://Your_Host:Your_HttpPort/jonasAdmin (Domain > Server JOnAs > Security).

For a **Datasource Security Realm**, the Process Console provides facilities to:

- Add or Remove users,
- Modify user profiles within the Bonita scope.

For managing users with an **LDAP directory**, only the association between users and profiles is modified through the Workflow Console. Adding or deleting Users is done according to a user-specific **LDAP Administration process**.

10.4 Initiating a New Datasource Security Realm for Bonita Workflow

10.4.1 Create the database

To create a new database, follow the specific **Database Administration process for that database**.

When using HSQL, the database is automatically created (if it does not already exist) when launching the JOnAS application server.

10.4.2 Adapt the JOnAS Datasource Configuration Files

The recommended way to do this is to keep the default Datasource names in the jonas.properties file (**bonita** and **jiapadmin**), and modify the properties files (bonita.properties and jonasadmin.properties) according to the new configuration. (That is, changing the URL to use a new database, and modify the Mapper and Driver to use a new rdbms).



Note:

It is mandatory to keep the JNDI names of these Datasources (bonita and jiapadmin, as referenced in the properties files).

The following are examples of the properties files for a PostgreSQL database, named MyDB.

These files are located under \$JONAS_BASE/conf.

Excerpt of bonita.properties file:

```
datasource.name      bonita
datasource.url       jdbc:postgresql://localhost:5432/db_jiapadmin211
datasource.classname org.postgresql.Driver
datasource.mapper    rdb.postgres
```

Excerpt of jonasadmin.properties file:

```
datasource.name=jiap
datasource.url=jdbc:postgresql://localhost:5432/db_jiapadmin211
datasource.mapper=rdb.postgres
datasource.classname=org.postgresql.Driver
```

Then stop and restart the application server with "Start JOnAS". Remember to copy the correct drivers under the \$JONAS_ROOT/lib/ext directory. These drivers are located under the \$BONITA_HOME/lib/ext directory.

Initialize the database(s):

Bonita Datasource: go to the BONITA_HOME directory, and execute the command:

ant initDB

Jiapadmin Datasource: go to the Bonita admin directory and execute:

ant -f initJiapDb.xml initJiapDb.

The databases are populated with the Bonita and jiapadmin tables, and the minimum required data.

**Note:**

Even if a specific Datasource security realm is already being used, the corresponding database must be initialized as described above. Only data contained in Bonita and jiapadmin tables are available for Bonita Workflow user management.

10.5 Configuring the LDAP Directory for Bonita Workflow-LoginModule Feature

10.5.1 JOnAS Configuration

Jonas-realm.xml file (path: Bonita-Workflow Installation directory \config\bonita):

Describe the LDAP realm for JOnAS. To do this, modify the jonas-realm.xml file (jonas-ldaprealm target) adding the LDAP entry point.

The following is an example of the added lines:

```
<ldaprealm name="ldaprlm_1"
  baseDN="dc=frec,dc=bull,dc=fr"
  initialContextFactory="com.sun.jndi.ldap.LdapCtxFactory"
  providerUrl="ldap://localhost:389"
  securityAuthentication="simple"
  securityPrincipal="cn=admin,dc=frec,dc=bull,dc=fr"
  securityCredentials="xxxxxx"
  authenticationMode="bind"
  userPasswordAttribute="userPassword"
  userRolesAttribute="memberOf"
  roleNameAttribute="cn"
  userDN="ou=bsoa_user"
  userSearchFilter="uid={0}"
  roleDN="ou=bsoa_group"
  roleSearchFilter="uniqueMember={0}"
  referral="throw" />
```

server.xml file (path: Bonita-Workflow Installation directory \config\bonita):

Modify the server.xml file to take this new realm into account.

First, modify the global realm:

```
Realm className="org.objectweb.jonas.security.realm.web.catalina50.JAAS"
debug="99" resourceName="Ldaprlm_1"/>
```



Note:

The Single Sign On (SSO) of Tomcat is activated.

Jaas.config file (path: Bonita-Workflow Installation directory \config\bonita):

To run the Bonita Workflow tests or Bonita Workflow samples mentioned in the Bonita Workflow documentation, remember to modify the jaas.config file specifying to jaas which security resource to use (modify all file entries).

Refer to the [JOnAS User Documentation](#) for farther information about modifying server.xml and jonas-realm.xml files to introduce a new security realm.

10.5.2 LDAP Configuration



Note:

To introduce valid information in the LDAP directory requires specific (password-protected) access with permission to modify the subTree roleDN.

Create a subtree containing the following groups:

```
JIAPUSER
JIAOPERATOR
JIAPDESIGNER
JIAPADMINISTRATOR
BONITAUSER
jonas_admin
```

These groups are mandatory for Bonita Workflow, except when using the BsoaLogin Module as explained at the end of this section. (JIAPUSER, BONITAUSER, jonas_admin are not required by the LDAP server).

JIAPUSER, JIAOPERATOR, JIAPDESIGNER, JIAPADMINISTRATOR are managed by the **Workflow Process Console**.

BONITAUSER and **jonas_admin** are managed by the **directory server administrator**.

The following is an example of the minimum configuration required to import into LDAP, based on the previously-mentioned LDAP:

```
dn: dc=frec,dc=bull,dc=fr
objectClass: top
objectClass: dcObject
objectClass: organization
o: bull
dc: frec

dn: cn=admin,dc=frec,dc=bull,dc=fr
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

dn: ou=bsoa_group,dc=frec,dc=bull,dc=fr
ou: bsoa_group
objectClass: top
objectClass: organizationalUnit

dn: ou=bsoa_user,dc=frec,dc=bull,dc=fr
ou: bsoa_user
objectClass: top
objectClass: organizationalUnit

dn: cn=JIAPUSER,ou=jiapadmin_group,dc=frec,dc=bull,dc=fr
cn: JIAPUSER
uniqueMember: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr
description: jiap user
objectClass: groupOfUniqueNames
objectClass: top

dn: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr
cn: jiap
sn: jiap
uid: jiap
objectClass: inetOrgPerson
objectClass: top

dn: cn=JIAPOPERATOR,ou=bsoa_group,dc=frec,dc=bull,dc=fr
cn: JIAPOPERATOR
uniqueMember: uid=jiap,dc=frec,dc=bull,dc=fr
objectClass: groupOfUniqueNames
objectClass: top

dn: cn=JIAPADMINISTRATOR,ou=bsoa_group,dc=frec,dc=bull,dc=fr
cn: JIAPADMINISTRATOR
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr
```

```

dn: cn=JIAPDESIGNER,ou=bsoa_group,dc=frec,dc=bull,dc=fr
cn: JIAPDESIGNER
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr

dn: cn=jonas_admin,ou=bsoa_group,dc=frec,dc=bull,dc=fr
cn: jonas_admin
uniqueMember: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr
description: jonas administration group
objectClass: groupOfUniqueNames
objectClass: top

dn: cn=BONITAUSER,ou=bsoa_group,dc=frec,dc=bull,dc=fr
cn: BONITAUSER
description: Jiap Acces Group
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=jiap,ou=bsoa_user,dc=frec,dc=bull,dc=fr

```



Note:

These entries are mandatory for using Bonita Workflow. To use Bonita, Bonita users must also be added in the LDAP Directory and assigned the “BONITAUSER” role.

After this is complete, restart the JOnAS server (bsoa/jonas start).

10.5.3 Using the jonasAdmin Tool

To access the JOnAS Administration Console, connect to the following URL

http://Your_Host:Your_HttpPort/jonasAdmin

To have access to this Administration console, the user must be a member of the “jonas_admin” group.

This administration console permits management of security realms. Instead of describing the LDAP realm in the jonas-realm.xml file as described previously in this section, initial configuration or modifications can be performed using the Administrative console.

Select the following path in the **Navigational Tree** (Left Panel) of the JOnAS console: **Domain(jonas) → Server JOnAS(jonas) → Security**

The right panel displays available LDAP Realms for modification or allows creation of new realms.

10.5.4 LoginModule Feature

This feature enables the administrator to automatically grant the authenticated user the BONITAUSER role and the JIAPUSER profile. Furthermore, if this user has a JIAPADMINISTRATOR profile, it automatically grants this user the "jonas_admin" role. This functionality prohibits managing previous groups from the LDAP server, but does allow access to the Workflow console granted to all users able to authenticate to the LDAP realm.

Jonas Configuration:

server.xml

```
<Realm className="org.objectweb.jonas.security.realm.web.catalina50.JAAS"
debug="99" resourceName="ldapr1m_1"/>
```

jaas.config

```
tomcat {
    org.objectweb.jonas.security.auth.spi.JResourceLoginModule required
    resourceName="ldapr1m_1";
auth.BsoaLoginModule required;
};
```

In the tomcat entry of the jaas.config file, add a line with "BsoaLoginModule" as specified above.

```
proed {
    org.objectweb.jonas.security.auth.spi.JResourceLoginModule required
    resourceName="ldapr1m_1";

    org.objectweb.jonas.security.auth.spi.ClientLoginModule required

    globalCtx="true"
auth.BsoaLoginModule required;
};
```

10.6 Selecting a New Security Realm for User Management

After the previous steps have been completed, configure Bonita Workflow modifying the User Management configuration.

Bonita Workflow administration

Connect to the Administration console with the appropriate user. (This can only be done with an administrator profile. If the basic installation was not changed, the default login/password: **bsoa/bsoa**) can be used.

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **User Management**. The **Core Frame** (Right Panel) displays the selected and available realms.

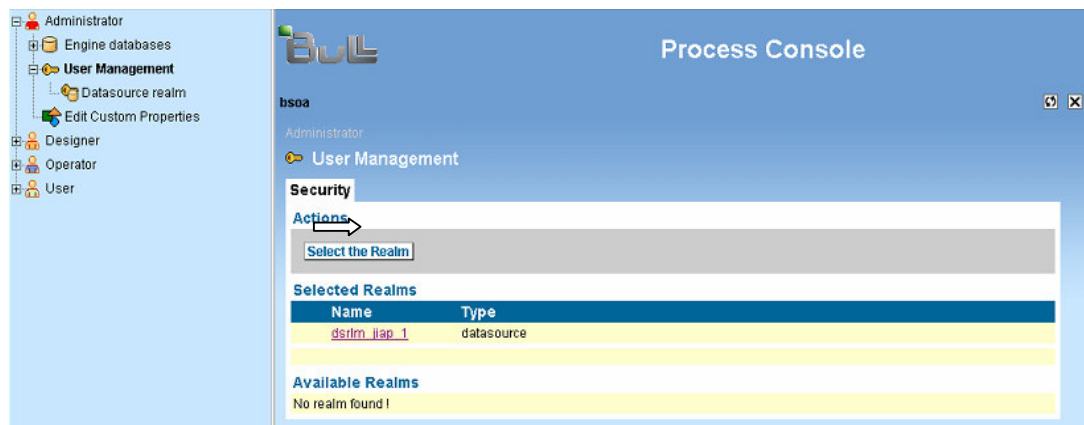


Figure 10-4. Datasource Realms Display in the Administration Console Core Frame

To select a new realm, check the box in front of the appropriate realm in the « Available Realms » list in the **Core Frame**, and then click on the « **Select the Realm** » Button.

A warning is then provided. Click the “**Confirm**” Button to confirm the choice.

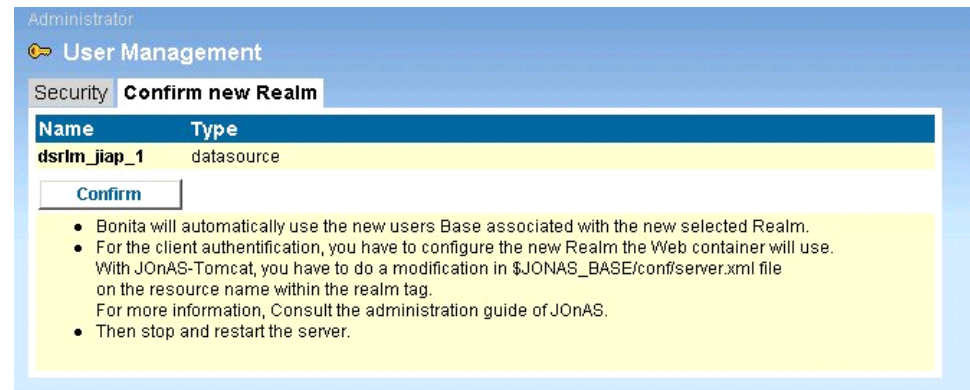


Figure 10-5. Confirmation Screen for Selecting a New Realm

Stop and restart the JOnAS Server. The new User Management configuration is now available.



Warning:

The modifications are immediately applied by the User Manager API within Bonita Workflow. This can result in problems when running applications. Stopping and restarting the JOnAS server is strongly recommended.

10.7 Accessing the description of a specific Realm

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **User Management**. The **Core Frame** (Right Panel) displays the available realms.

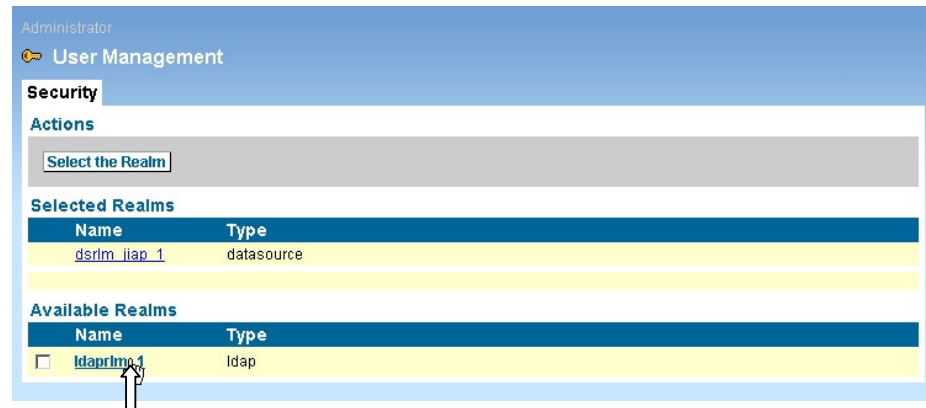


Figure 10-6. Display of Available Realms

Click on the name of the desired Realm and related information is displayed in the **Core Frame**, as shown below for an LDAP Realm.

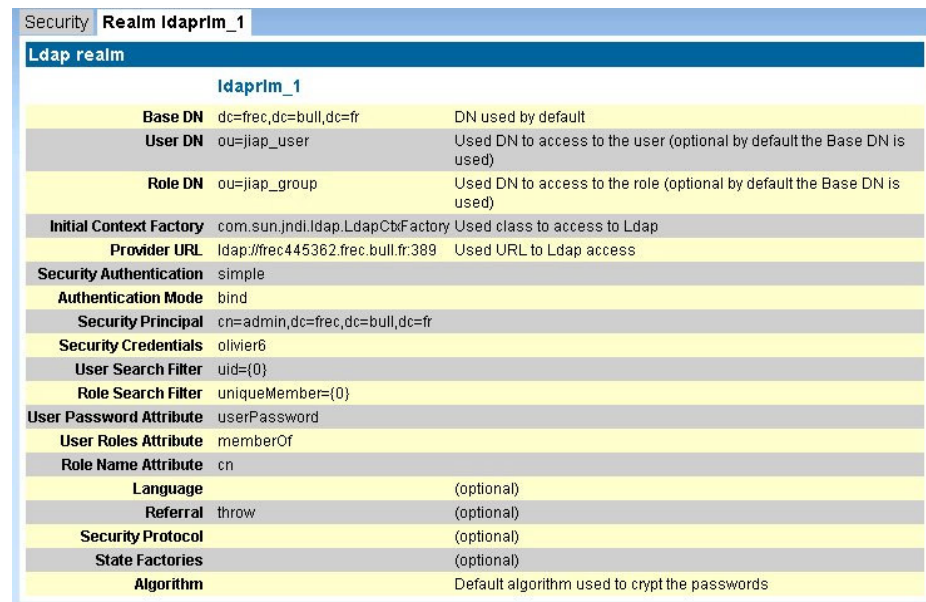


Figure 10-7. Detailed Information for an LDAP Realm

10.8 Accessing the Users List of a Specific Realm



Warning:

This operation is only performed on the selected security realm. (See the previous section for more information about how to select a specific Security Realm).

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **User Management**. The **Core Frame** (Right Panel) displays the selected realm.

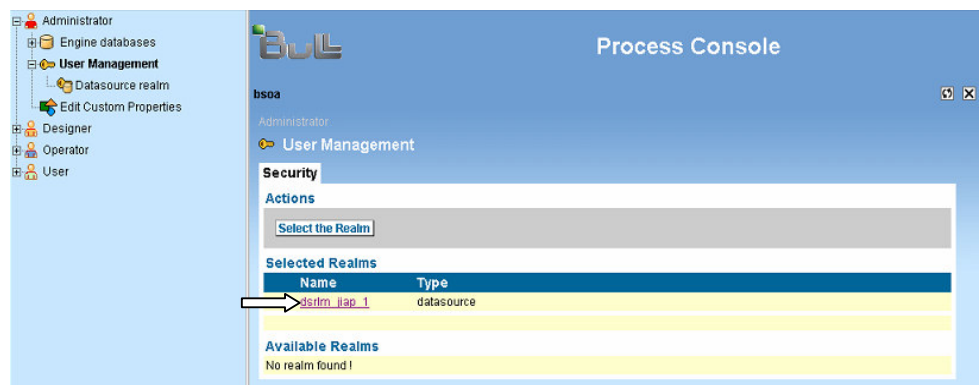


Figure 10-8. Display of Selected Realm

Click on the name of the selected Realm, choose the “Users” tab in the **Core Frame**, and click on the “**SEARCH**” Button. The list of users is displayed under the search panel.

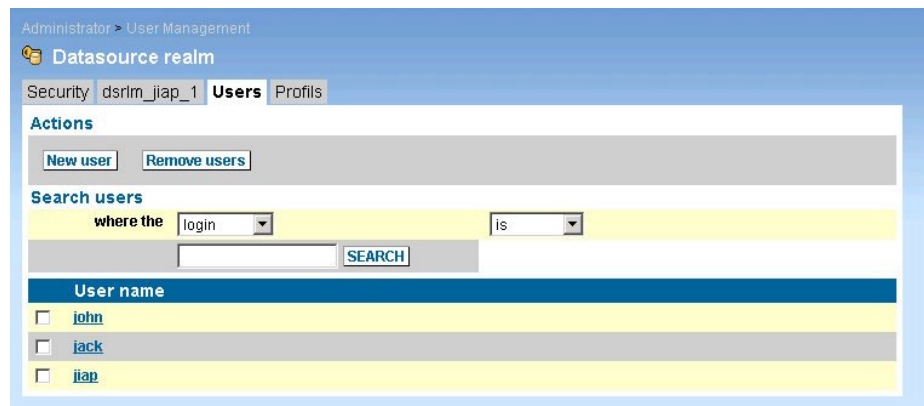


Figure 10-9. List of Users for a Selected Realm

10.9 Adding a User To a Specific Realm



Warning:

This operation is only performed on the selected security realm. (See the previous sections for more information about selecting a specific Security Realm.)



Note:

Realms are managed with their specific administration tool.

First, access the list of users for the realm (see "Accessing the Users List of a Specific Realm"). Then click on the "**New user**" Button. The user registration form is displayed in the **Core Frame**.

Figure 10-10. New User Registration Form

Enter the user information and click on the "**Create**" button.

For more information about the "Profiles" panel, see the "Accessing the List of Users Involved in a Specific Bonita Profile" section.

The user is added to the security realm.

10.10 Deleting a User From a Specific Realm



Warning:

This operation is performed only on the selected security realm. (See the previous sections for more information on how to select a specific Security Realm.)



Note:

LDAP Realms are managed with their specific administration tool.

First, access the user list of the realm (see "Accessing the Users List of a Specific Realm"). Then check the box in front of the name of the user(s) to delete, and click on the "**Remove Users**" Button.

Administrator > User Management

Datasource realm

Security dsrlm_jiap_1 **Users** Profiles

Actions

New user Remove users

Search users

where the login is

SEARCH

User name	
<input type="checkbox"/>	john
<input type="checkbox"/>	jack
<input checked="" type="checkbox"/>	chris
<input type="checkbox"/>	jiap

Figure 10-11. Form for Removing Users from a Specific Realm

The selected users are deleted from the list (replay **SEARCH** to confirm the change).



Caution:

These deletions are effective immediately. This could result in problems for users involved in a current Workflow process.

10.11 Accessing the List of Users Involved in a Specific Bonita Profile

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **User Management**. The **Core Frame** (Right Panel) displays the selected realm.

Click on the name of the selected Realm, and choose the “**Profiles**” tab in the **Core Frame**.

The four profiles are shown: select one of them by clicking on it.

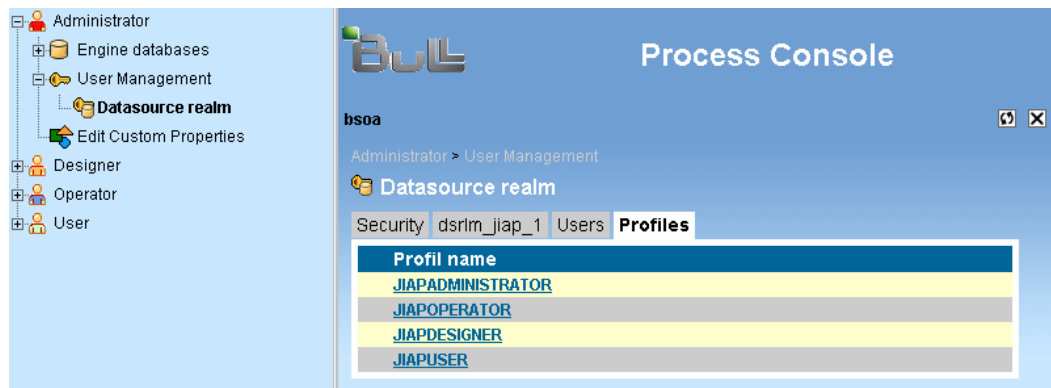


Figure 10-12. List of Available Profiles

A new tab is available in the Core Frame, listing the users involved in the selected profile.

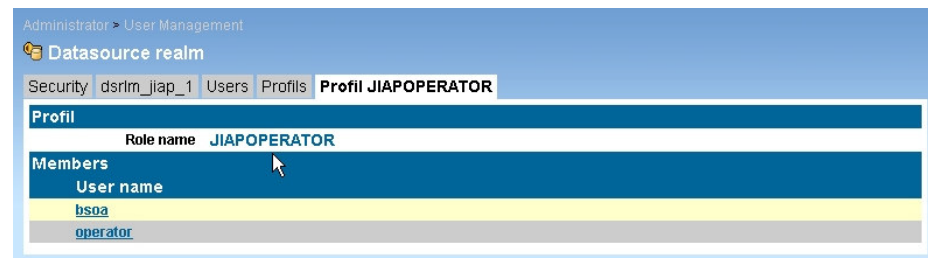




Figure 10-13. List of Users for a Specific Profile

To add a user to this profile:

Select the user in the list by clicking on the name. The “**User xxx**” tab is displayed in the **Core Frame**. Then add the chosen profile for this user by first clicking on it in the “**Available**” panel, then clicking on the  arrow.

To delete a user from this profile:

Select the user by clicking on the name. The “**User xxx**” tab is displayed in the **Core Frame**. Then delete the chosen profile from this list by first clicking on it in the **User's profile** panel, then clicking on the  arrow.

10.12 Modifying Bonita Profiles or Passwords for a Specific User



Warning:


This operation is performed only on the selected security realm. (See the previous sections for more information on selecting a specific Security Realm) in case of a Datasource Realm.

First access the users list of the realm (see "Accessing the Users List of a Specific Realm"), and then select a user by clicking on the name.


The user information form is shown in the **Core Frame**.

Figure 10-14. User Information Form

To add a profile for this user:

Select the profile by clicking on it in the **Available** panel (on the right), then click on the  arrow. The profile is displayed in the **User's profile** panel.

To delete a profile for this user:

Select the profile by clicking on it in the **User's profile** panel, then click on the  arrow. The profile is deleted from the user's profiles.

For other changes, modify the user information as required.

To validate the changes, click on the “**Apply**” button when finished.



Caution:

These changes are effective immediately. Changing groups could result in problems for users currently running the Process Console.

Chapter 11. Engine Databases

Within this version of Bonita Workflow, only the Bonita Engine is provided. In subsequent versions, an orchestration engine will also be included.

The Process console offers a means for accessing information about the current Datasource configuration.

11.1 Changing the Basic Engine Datasource

Specifying Datasources (datasource mapper, datasource url, ...), is not in the scope of the Process Console.

This is done at installation time and is described in the **Bonita Installation Guide**. The Bonita Installation Guide explains how to adapt `jonas.properties` and `[datasourceName].xml` files.

To access the Datasource configuration and to modify the JDBC connection parameters, use the Jonas admin console `http://Your_Host:Your_HttpPort/jonasAdmin` (Domain → Server JOnAS → Services → Database).

11.2 Displaying the Engine Datasource Definition

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **Engine Databases** → **Workflow Database**.

The Bonita Datasource configuration is displayed in the **Core Frame** (Right Panel).

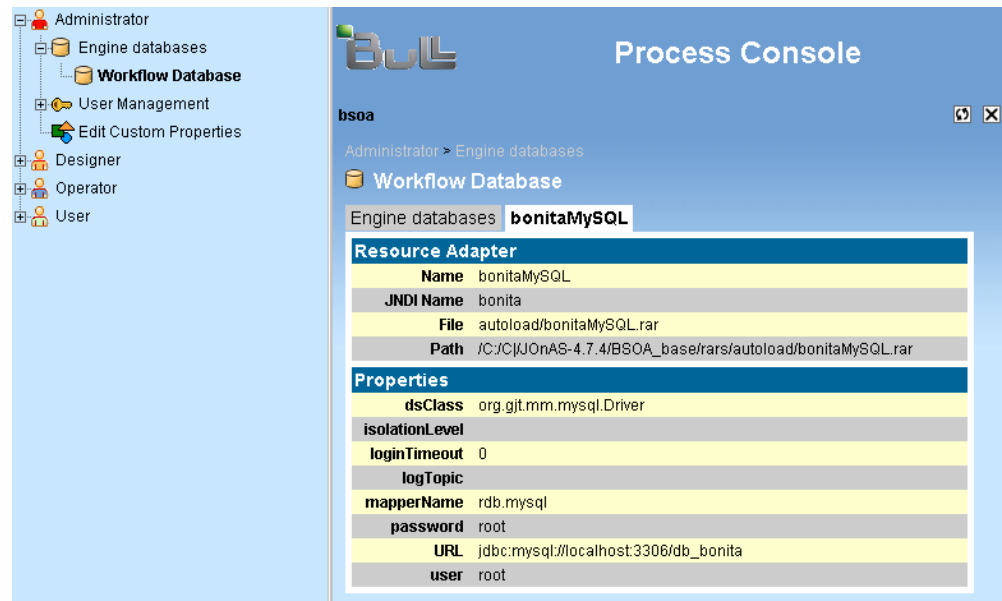
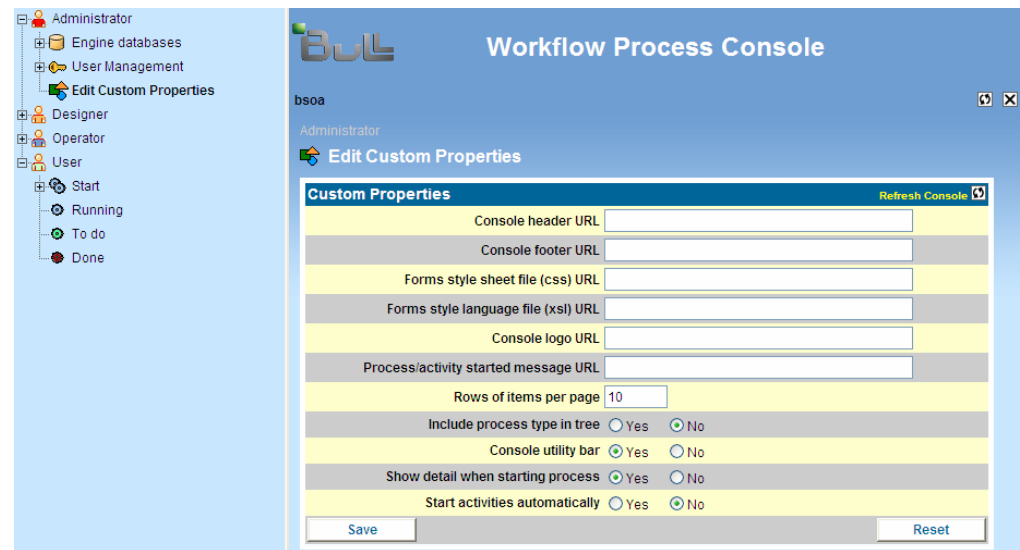


Figure 11-1. Bonita Datasource Configuration

Chapter 12. Editing Custom Properties

Select the following path in the **Navigational Tree** (Left Panel): **Administrator** → **Edit Custom Properties**.

The Custom Properties configuration is displayed in the **Core Frame** (Right Panel).




The screenshot displays the 'Workflow Process Console' interface. On the left is the 'Navigational Tree' with a tree structure: Administrator (selected), Engine databases, User Management, Edit Custom Properties, Designer, Operator, User, Start, Running, To do, and Done. The main area on the right is titled 'Workflow Process Console' and shows the 'Edit Custom Properties' configuration. It includes a 'Refresh Console' button and a list of properties with input fields and radio buttons. The properties are: Console header URL, Console footer URL, Forms style sheet file (css) URL, Forms style language file (xsl) URL, Console logo URL, Process/activity started message URL, Rows of items per page (set to 10), Include process type in tree (radio buttons for Yes and No, with No selected), Console utility bar (radio buttons for Yes and No, with Yes selected), Show detail when starting process (radio buttons for Yes and No, with Yes selected), and Start activities automatically (radio buttons for Yes and No, with No selected). At the bottom are 'Save' and 'Reset' buttons.

Custom Properties		Refresh Console
Console header URL	<input type="text"/>	
Console footer URL	<input type="text"/>	
Forms style sheet file (css) URL	<input type="text"/>	
Forms style language file (xsl) URL	<input type="text"/>	
Console logo URL	<input type="text"/>	
Process/activity started message URL	<input type="text"/>	
Rows of items per page	<input type="text" value="10"/>	
Include process type in tree	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Console utility bar	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Show detail when starting process	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Start activities automatically	<input type="radio"/> Yes <input checked="" type="radio"/> No	
<input type="button" value="Save"/>		<input type="button" value="Reset"/>

Figure 12-1. Custom Properties Configuration Screen

Property customization is accomplished by inserting a URL that points to a customized file. The content provides HTML for display in the header or footer frames, a Cascading Style Sheet (CSS), or Extensible Style Language (XSL) definition file.

Once the URL(s) are inserted, click the **Save** button. The administrator is prompted to confirm the save action. Once confirmation is performed, refresh the console () to display the changes.

Chapter 13. Database Tuning

In order to improve the performance during the Workflow execution, it becomes necessary to modify the default database schema generated by the JOnAS Application Server.

In fact, this schema does NOT contain either the foreign key or the indexes definitions; thus, this will significantly impact the entire system performance. The following lines explain the necessary steps to update the default-generated schema.

Just after the Bonita Workflow start operation (via the graphical shortcuts for the windows version or the jonas start operation on Linux) the JOnAS server will create the default database schema according to a selected relational database. The following is the default schema created for the HSQL database (default database after Bonita Workflow installation):

```
CREATE MEMORY TABLE BNNEXTNUMBER(MAXNUMBER INTEGER, NAME VARCHAR, ID VARCHAR NOT
NULL PRIMARY KEY)

CREATE MEMORY TABLE BNNODEPERFORMERASSIGN(TYPE INTEGER, PROPERTY
VARCHAR, BNNODE_ID VARCHAR, NAME VARCHAR, ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNITERATION(FROMNODE VARCHAR, ITERATIONCONDITION VARCHAR, TONODE
VARCHAR, BNPROJECT_ID VARCHAR, ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNEDGE(FK_INNODE VARCHAR, BNPROJECT_ID VARCHAR, BNNODE_ID
VARCHAR, STATE INTEGER, NAME VARCHAR, ID VARCHAR NOT NULL PRIMARY
KEY, EDGECONDITION VARCHAR, CREATIONDATE TIMESTAMP, MODIFICATIONDATE TIMESTAMP)

CREATE MEMORY TABLE BNAUTHROLE(AUTHROLEGROUP VARCHAR, NAME VARCHAR, ID VARCHAR NOT
NULL PRIMARY KEY)

CREATE MEMORY TABLE BNAUTHROLE_BNUSER(BNAUTHROLE_ID VARCHAR, BNUSER_ID VARCHAR)

CREATE MEMORY TABLE BNROLEMAPPER(TYPE INTEGER, FK_ROLE VARCHAR, NAME VARCHAR, ID
VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNROLE(DESCRIPTION VARCHAR, BNPROJECT_ID VARCHAR, NAME
VARCHAR, ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNUSER_BNROLE(BNROLE_ID VARCHAR, BNUSER_ID VARCHAR)

CREATE MEMORY TABLE BNNODE(BNROLE_ID VARCHAR, ID VARCHAR NOT NULL PRIMARY
KEY, BNPROJECT_ID VARCHAR, STARTDATE TIMESTAMP, EXECUTOR VARCHAR, ANTICIPABLE
BOOLEAN, ENDDATE TIMESTAMP, TYPE INTEGER, ACTIVITYPERFORMER VARCHAR, DEADLINES
VARBINARY, RELATIVEDEADLINES VARBINARY, REFERENCE VARCHAR, CREATOR
VARCHAR, DESCRIPTION VARCHAR, STATE INTEGER, NAME VARCHAR, CREATIONDATE TIMESTAMP)
CREATE MEMORY TABLE BNNODEHOOK(TYPE INTEGER, EVENT VARCHAR, BNNODE_ID VARCHAR, NAME
VARCHAR, ID VARCHAR NOT NULL PRIMARY KEY)
CREATE MEMORY TABLE BNNODEPROPERTY(POSSIBLEVALUES VARBINARY, BNNODE_ID
VARCHAR, THEKEY VARCHAR, PROPAGATE BOOLEAN, ID VARCHAR NOT NULL PRIMARY KEY, THEVALUE
VARCHAR)

CREATE MEMORY TABLE BNNODEINTERHOOK(TYPE INTEGER, SCRIPT VARCHAR, EVENT
VARCHAR, BNNODE_ID VARCHAR, NAME VARCHAR, ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNUSERPROPERTY(THEKEY VARCHAR, BNUSER_ID VARCHAR, ID VARCHAR NOT
NULL PRIMARY KEY, THEVALUE VARCHAR)

CREATE MEMORY HOOK(TYPE INTEGER, BNPROJECT_ID VARCHAR, EVENT VARCHAR, NAME VARCHAR, ID
VARCHAR NOT NULL PRIMARY KEY)
```

```

CREATE MEMORY TABLE BNPROJECTPROPERTY(POSSIBLEVALUES VARBINARY,BNPROJECT_ID
VARCHAR,THEKEY VARCHAR,ID VARCHAR NOT NULL PRIMARY KEY,THEVALUE VARCHAR)

CREATE MEMORY TABLE BNINITIATORMAPPER(TYPE INTEGER,FK_PROJECT VARCHAR,NAME
VARCHAR,ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNPROJECTINTERHOOK(TYPE INTEGER,SCRIPT VARCHAR,BNPROJECT_ID
VARCHAR,EVENT VARCHAR,NAME VARCHAR,ID VARCHAR NOT NULL PRIMARY KEY)

CREATE MEMORY TABLE BNUSER(PASSWORD VARCHAR,EMAIL VARCHAR,NAME VARCHAR,JABBER
VARCHAR,ID VARCHAR NOT NULL PRIMARY KEY,CREATIONDATE TIMESTAMP,MODIFICATIONDATE
TIMESTAMP)

CREATE MEMORY TABLE BNUSER_BNPROJECT(BNPROJECT_ID VARCHAR,BNUSER_ID VARCHAR)

CREATE MEMORY TABLE BNAGENT(BNUSER_ID VARCHAR,ID VARCHAR NOT NULL PRIMARY
KEY,BNPROJECT_ID VARCHAR,DESCRIPTION VARCHAR,STATE INTEGER,NAME
VARCHAR,CREATIONDATE TIMESTAMP)

CREATE MEMORY TABLE BNPROJECT(CREATOR VARCHAR,ENDDATE TIMESTAMP,TYPE VARCHAR,STATE
INTEGER,STATUS VARCHAR,NAME VARCHAR,ID VARCHAR NOT NULL PRIMARY KEY,PARENT
VARCHAR,VERSION VARCHAR,CREATIONDATE TIMESTAMP)

CREATE MEMORY TABLE BNAGENTEDGE(BNPROJECT_ID VARCHAR,ID VARCHAR NOT NULL PRIMARY
KEY,BNNODE_ID VARCHAR,BNAGENT_ID VARCHAR,STATE INTEGER,NAME VARCHAR,CREATIONDATE
TIMESTAMP)

```

The previous scheme can be improved by adding the foreign keys and indexes for the most queried columns. As the default schema could change depending on the relational database used, it is recommended that the following commands be applied to the default schema just after it is created:

```

ALTER TABLE BnNodePerformerAssign ADD CONSTRAINT F_BnNodePerfAssign_id FOREIGN KEY
(BnNode_id) REFERENCES BnNode(id)

ALTER TABLE BnIteration ADD CONSTRAINT F_BnIteration_BnProject_id FOREIGN KEY
(BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnEdge ADD CONSTRAINT F_BnEdge_fk_innode FOREIGN KEY (fk_innode)
REFERENCES BnNode(id)

ALTER TABLE BnEdge ADD CONSTRAINT F_BnEdge_BnNode_id FOREIGN KEY (BnNode_id)
REFERENCES BnNode(id)

ALTER TABLE BnEdge ADD CONSTRAINT F_BnEdge_BnProject_id FOREIGN KEY (BnProject_id)
REFERENCES BnProject(id)

ALTER TABLE BnRoleMapper ADD CONSTRAINT F_BnRole_BnRoleMapper_id FOREIGN KEY
(fk_role) REFERENCES BnRole(id)

ALTER TABLE BnInitiatorMapper ADD CONSTRAINT F_BnProject_BnInitiatorMapper_id
FOREIGN KEY (fk_project) REFERENCES BnProject(id)

ALTER TABLE BnRole ADD CONSTRAINT F_BnRole_BnProject_id FOREIGN KEY (BnProject_id)
REFERENCES BnProject(id)

ALTER TABLE BnNode ADD CONSTRAINT F_BnNode_BnProject_id FOREIGN KEY (BnProject_id)
REFERENCES BnProject(id)

ALTER TABLE BnNode ADD CONSTRAINT F_BnNode_BnRole_id FOREIGN KEY (BnRole_id)
REFERENCES BnRole(id)

ALTER TABLE BnNodeHook ADD CONSTRAINT F_BnNodeHook_BnNode_id FOREIGN KEY
(BnNode_id) REFERENCES BnNode(id)

ALTER TABLE BnNodeProperty ADD CONSTRAINT F_BnNodeProperty_BnNode_id FOREIGN KEY
(BnNode_id) REFERENCES BnNode(id)

```



```

ALTER TABLE BnNodeInterHook ADD CONSTRAINT F_BnNodeInterHook_BnNode_id FOREIGN KEY
(BnNode_id) REFERENCES BnNode(id)

ALTER TABLE BnUserProperty ADD CONSTRAINT F_BnUserProperty_BnUser_id FOREIGN KEY
(BnUser_id) REFERENCES BnUser(id)

ALTER TABLE BnProjectHook ADD CONSTRAINT F_BnProjectHook_BnProject_id FOREIGN KEY
(BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnProjectProperty ADD CONSTRAINT F_BnProjectProp_BnProject_id FOREIGN
KEY (BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnProjectInterHook ADD CONSTRAINT F_BnProjectIH_BnProject_id FOREIGN
KEY (BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnAgent ADD CONSTRAINT F_BnAgent_BnUser_id FOREIGN KEY (BnUser_id)
REFERENCES BnUser(id)

ALTER TABLE BnAgent ADD CONSTRAINT F_BnAgent_BnProject_id FOREIGN KEY
(BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnAgentEdge ADD CONSTRAINT F_BnAgentEdge_BnProject_id FOREIGN KEY
(BnProject_id) REFERENCES BnProject(id)

ALTER TABLE BnAgentEdge ADD CONSTRAINT F_BnAgentEdge_BnNode_id FOREIGN KEY
(BnNode_id) REFERENCES BnNode(id)

ALTER TABLE BnAgentEdge ADD CONSTRAINT F_BnAgentEdge_BnAgent_id FOREIGN KEY
(BnAgent_id) REFERENCES BnAgent(id)

ALTER TABLE BNAUTHROLE_BNUSER ADD CONSTRAINT F_BNAUTHROLE_BNUSER_AuthR_id FOREIGN
KEY (BnAuthRole_id) REFERENCES BnAuthRole(id) ON DELETE CASCADE

ALTER TABLE BNAUTHROLE_BNUSER ADD CONSTRAINT F_BNAUTHROLE_BNUSER_User_id FOREIGN
KEY (BnUser_id) REFERENCES BnUser(id) ON DELETE CASCADE

ALTER TABLE BNUSER_BNROLE ADD CONSTRAINT F_BNUSER_BNROLE_BnRole_id FOREIGN KEY
(BnRole_id) REFERENCES BnRole(id) ON DELETE CASCADE

ALTER TABLE BNUSER_BNROLE ADD CONSTRAINT F_BNUSER_BNROLE_BnUser_id FOREIGN KEY
(BnUser_id) REFERENCES BnUser(id) ON DELETE CASCADE

ALTER TABLE BNUSER_BNPROJECT ADD CONSTRAINT F_BNUSER_BNPROJECT_Project_id FOREIGN
KEY (BnProject_id) REFERENCES BnProject(id) ON DELETE CASCADE

ALTER TABLE BNUSER_BNPROJECT ADD CONSTRAINT F_BNUSER_BNPROJECT_BnUser_id FOREIGN
KEY (BnUser_id) REFERENCES BnUser(id) ON DELETE CASCADE

```

Once this operation is finished, you could proceed with the following columns indexations commands:

```

CREATE INDEX I_BnNodePerfAssign_BnNode_id ON BnNodePerformerAssign (BnNode_id);

CREATE INDEX I_BnIteration_BnProject_id ON BnIteration (BnProject_id );

CREATE INDEX I_BnEdge_fk_innode ON BnEdge (fk_innode);

CREATE INDEX I_BnEdge_BnNode_id ON BnEdge (BnNode_id);

CREATE INDEX I_BnEdge_BnProject_id ON BnEdge (BnProject_id);

CREATE INDEX I_BnRoleMapper_fk_role ON BnRoleMapper (fk_role);

CREATE INDEX I_BnRole_BnProject_id ON BnRole (BnProject_id);

CREATE INDEX I_BnNode_BnProject_id ON BnNode (BnProject_id);

CREATE INDEX I_BnNode_BnRole_id ON BnNode (BnRole_id);

CREATE INDEX I_BnNodeHook_BnNode_id ON BnNodeHook (BnNode_id);

```

```

CREATE INDEX I_BnNodeProperty_BnNode_id ON BnNodeProperty (BnNode_id);
CREATE INDEX I_BnNodeInterHook_BnNode_id ON BnNodeInterHook (BnNode_id);
CREATE INDEX I_BnUserProperty_BnUser_id ON BnUserProperty (BnUser_id);
CREATE INDEX I_BnProjectHook_BnProject_id ON BnProjectHook (BnProject_id);
CREATE INDEX I_BnProjectProp_BnProject_id ON BnProjectProperty (BnProject_id);
CREATE INDEX I_BnProjectIH_BnProject_id ON BnProjectInterHook (BnProject_id);
CREATE INDEX I_BnAgent_BnUser_id ON BnAgent (BnUser_id);
CREATE INDEX I_BnAgent_BnProject_id ON BnAgent (BnProject_id);
CREATE INDEX I_BnAgentEdge_BnProject_id ON BnAgentEdge (BnProject_id);
CREATE INDEX I_BnAgentEdge_BnNode_id ON BnAgentEdge (BnNode_id);
CREATE INDEX I_BnAgentEdge_BnAgent_id ON BnAgentEdge (BnAgent_id);
CREATE INDEX I_BNAUTHROLE_BNUSER_AuthR_id ON BNAUTHROLE_BNUSER (BnAuthRole_id);
CREATE INDEX I_BNAUTHROLE_BNUSER_User_id ON BNAUTHROLE_BNUSER (BnUser_id);
CREATE INDEX I_BNUSER_BNROLE_BnRole_id ON BNUSER_BNROLE (BnRole_id);
CREATE INDEX I_BNUSER_BNROLE_BnUser_id ON BNUSER_BNROLE (BnUser_id);
CREATE INDEX I_BNUSER_BNPROJECT_Project_id ON BNUSER_BNPROJECT (BnProject_id);
CREATE INDEX I_BNUSER_BNPROJECT_BnUser_id ON BNUSER_BNPROJECT (BnUser_id);
CREATE INDEX I_BnProject_Name on BnProject(name);
CREATE INDEX I_BnUser_Name on BnUser(name);
CREATE INDEX I_BnNode_Name on BnNode(name);
CREATE INDEX I_BnNode_ActivityPerformer on BnNode(activityPerformer);
CREATE INDEX I_BnProjectProperty_thekey on BnProjectProperty(thekey);
CREATE INDEX I_BnNodeProperty_thekey on BnNodeProperty(thekey);
CREATE INDEX I_BnNode_state on BnNode(state);
CREATE INDEX I_BnRole_name on BnRole(name);
CREATE INDEX I_BnIM_BnProject_id ON BnInitiatorMapper (fk_project);

```

Once both operations are completed, it is necessary to stop and restart the Bonita Workflow system.

Chapter 14. Database Migration Considerations

In order to run the Bonita Workflow v3.0 with an existing database that was created with Bonita Workflow v2.5.x, apply the following commands.



CAUTION:

For projects having sub processes all instances of these projects must be terminated and then archived or purged from the database.

```
ALTER TABLE BnNode ADD reference VARCHAR(256) ;
ALTER TABLE BnNode DROP COLUMN transition ;
ALTER TABLE BnNode DROP COLUMN activation ;
ALTER TABLE BnProject ADD version VARCHAR(256) ;
UPDATE BnProject
SET version = '1.0';
UPDATE NnNode
SET reference=name || '_version1.0'
WHERE type='5';
```



Important:

Due to the new versioning feature introduced with Bonita Workflow v3.0, Workflow assumes that for all projects created with Bonita Workflow v2.5, the version column is set to 1.0 within the database.

