

BioMAJ Watcher User Guide

Table of contents

1. Application purpose	3
2. Features.....	3
3. Setting up	3
3.1. Configuration	3
3.1.1. Authentication	3
3.1.2. Admin selection	4
3.1.3. Host selection.....	4
3.1.4. Other parameters	4
3.2. Start up.....	5
3.2.1. Embedded tomcat server	5
3.2.2. Already installed server	5
4. Multitenancy	5
4.1. Customized view	5
4.2. Override general settings.....	5
5. User interface.....	5
5.1. Main interface.....	6
5.2. User management.....	8
5.3. Bank properties edition	9
5.3.1. Main panel	9
5.3.2. Process edition panel.....	11
5.3.3. Site manager	12
5.3.4. Properties selection panel	12
6. Running bank updates	13
6.1. Manual updates	13
6.2. Scheduled updates.....	13
7. REST Querying service.....	13
7.1. Retrieving bank information	13
7.1.1. The formats list	13
7.1.2. The types list	14
7.1.3. The banks list	14
7.2. Downloading files.....	16
7.3. Starting updates.....	17

Table of figures

Figure 1: BW main interface	6
Figure 2: Bank ownership color	7
Figure 3: Admin panel	8
Figure 4: BW bank edition.....	9
Figure 5: Process visualization	11
Figure 6: Remote repositories manager	12
Figure 7: Properties selection panel	12

1. Application purpose

Biomaj Watcher (BW) has been designed to provide a quick overview of the banks managed by Biomaj and a simple access to the most common administration tasks.

2. Features

BW allows :

- A guest user to visualize/search/filter the online banks. It includes charts and detailed information about a bank.
- A logged-in user to :
 - Run updates
 - Visualize the status of the banks
 - Schedule updates with a cron-like syntax.
 - View the latest errors with access to the related log files
 - Create/modify banks properties

3. Setting up

Most of the configuration for BW is done via the installer, but you can manually modify it via the properties in, preferably:

```
<tomcat dir>/conf/Catalina/<host>/BmajWatcher.xml
```

Or:

```
<tomcat dir>/webapps/BmajWatcher/WEB-INF/web.xml
```

3.1. Configuration

3.1.1. Authentication

Two authentication methods are available: local authentication and LDAP authentication.

3.1.1.1. *Local authentication*

Local authentication is done against the Biomaj database. You need to have at least one user present in the database (should have been added during Biomaj installation).

To enable local authentication:

- Set the USELOCAL property to 1.

3.1.1.2. *LDAP authentication*

To enable LDAP authentication, modify the following parameters in the configuration file:

- USELDAP : Set to 1.
- LDAPHOST : LDAP server address
- LDAPDN : LDAP distinguished name
- OPTFILTER : Search filter

3.1.2. Administrator selection

The database holds no information about which user is the administrator. The admin selection is done in the configuration file by specifying what user (identified by his login) should be given the admin privileges.

- ADMIN_LOGIN parameter set to desired login.

3.1.3. Host selection

3.1.3.1. Local host

The default behavior for BW is to run the bank updates on the localhost. It means that you need both BW and BioMAJ to be installed on the same computer. It is set with the parameter EXECUTION set to local.

3.1.3.2. Remote host

BW can run updates on any host that is reachable via SSH. However, there are some requirements to use this functionality:

- BW does not handle the update process. It only starts "biomaj.sh" in an external shell with the required parameters. So the remote servers must have BioMAJ installed.
- BW and BioMAJ access common data such as the banks properties files or the production directories. Therefore each host must share the same data and configuration. More precisely, the following directories have to be shared:
 - BIOMAJ_ROOT directory and its subdirectories (bin, log, tmp, conf...) defined in \$BIOMAJ_ROOT/general.conf.
 - data.dir that contains the production directories

Remote execution is configured with the following parameters:

- EXECUTION : set to remote
- SSH_AUTH_TYPE: login for login/password authentication or key for SSH key authentication.
- SSH_LOGIN: user login
- SSH_PASSWD: user password
- SSH_KEY_PATH: private key location for key authentication
- SSH_PASSPHRASE: passphrase for key authentication
- SSH_HOSTS: hosts list to connect to, separated by a comma.

If several hosts are specified:

- BW will connect to one that is available: BW considers a host to be available if an update is not already running on that host. If no hosts are available, BW will connect to the next one in its list.
- If a connection to a specific host fails, BW will try to connect to each remaining host until it succeeds.

3.1.4. Other parameters

- BIOMAJ_ROOT: Main Biomaj installation directory. Same as \$BIOMAJ_ROOT
- JOBS_LOCATION: Absolute path of your jobs.xml/jobs.xsd files.
- MESSAGE_MAX: Maximum number of error messages to display in the "errors" tab of the web interface.

3.2.Start up

3.2.1. Embedded tomcat server

To run BW, start the application server that hosts it. If it is the tomcat server shipped with the biomaj-<version>.tar.gz archive, run `apache-tomcat-6.0.24/bin/startup.sh`. The application can be reached at `http://<server>:8080/BmajWatcher/`

3.2.2. Already installed server

If you already have a tomcat server running, copy the following files in your tomcat directory:

- `<tomcat>/webapps/BmajWatcher.war`
- `<tomcat>/lib/biomaj.jar` (symbolic link)
- `<tomcat>conf/Catalina/<host>/BmajWatcher.xml`
- `<tomcat>conf/Catalina/<host>/BmajWatcher#logs.xml`

4. Multitenancy

4.1.Customized view

BW supports multitenancy, meaning that each user sees a different content depending on his privileges on the banks. It has introduced user group and bank ownership notions.

What a user has access to is detailed in the following table:

	Private bank	Public bank
Belongs to user	Full access	Full access
Belongs to a group member	Full access	Full access
Belongs to another group	No access	Restricted access

The default access for a bank is defined in “global.properties” with the property:
`visibility.default=private|public`

Looking into the consequences of multitenancy on the interface:

- Scheduler: The user only sees the jobs of his group
- Errors: The user only sees the errors of his group
- Global statistics: Only statistics of the group. If not logged, the statistics are processed for the public banks.

4.2.Override general settings

Each user inherits “global.properties” but only the admin can modify it.

If you have specific settings, make sure you override the required properties. You have 2 ways to do so:

- Rewrite the properties in the bank file.
- Rewrite the properties in an intermediate file that you include in your bank file. This way it is reusable.



If you override `data.dir`, make sure that the user that runs tomcat has write access on the directory.

5. User interface

5.1.Main interface

BioMAJ Watcher

The screenshot displays the BioMAJ Watcher main interface. At the top, there are tabs for 'Banks Status', 'Statistics', and 'Latest errors'. A dropdown menu is set to 'Update Drosophila_pseudobscura-flybase'. A toolbar contains icons for 'Bank detail', 'Show task', and 'New bank'. A table lists various banks with columns for 'Sched', 'Type', 'Formats', 'Name', 'Release', 'Latest session date', 'Status', and 'Access'. A context menu is open over a row, showing options like 'Change visibility', 'Change bank owner', 'Edit bank properties', 'Show task', 'Start update', 'Start update from scratch', 'New update (--new)', 'Rebuild bank (--rebuild)', and 'Remove'. A 'Scheduler' window is visible, showing a list of tasks with a dropdown for '7,14,21,28,31 du mois'. A 'Login' window shows 'Welcome admin' and a 'Logout' button. A 'Bank overview' window for 'gag' is open, showing a dependency tree with 'gag' at the top, branching into 'gag_ensembl', 'ncbi_gene', and 'hgnc'. Below this, 'Production directories' are listed with timestamps and paths. Other sections include 'Properties', 'Dependencies', 'Release', 'Pre-processes', and 'Post-processes'. A 'Close' button is at the bottom right of the 'Bank overview' window.

Numbered callouts in the image:

- 1: Context menu
- 2: Bank overview window tabs
- 3: Dependency tree diagram
- 4: Scheduler window
- 5: Scheduler task dropdown
- 6: Scheduler task icon
- 7: 'Show task' menu item
- 8: 'New bank' toolbar icon
- 9: 'Statistics' tab
- 10: Scheduler task status icon
- 11: 'Welcome admin' login message

Figure 1: BW main interface

- 1: Grid that contains the banks
- 2: Detailed information on a bank that shows up when user double clicks on a bank
- 3: Bank dependencies visualization
- 4: Scheduler where the jobs are displayed
- 5: A specific job. To remove a bank from a job, right click on the bank name then click on 'remove'. Banks are added to a job by drag&dropping them from the main grid to the desired task.
 - First icon is for updating immediately all the bank of this schedule
 - Second icon is for editing the schedule
 - Third icon is for deleting the schedule
- 6: The first button is for creating a new job. The second button is to display a calendar view of the jobs.
- 7: The contextual menu that offers several options for the selected bank.
- 8: The toolbar that allows you, in addition to the options of the contextual menu, to display the banks type hierarchy.
- 9: The other available tabs: 'Statistics' provides general statistics for the banks. 'Latest errors' displays the latest errors/warning for the updated banks.
- 10: Refresh the scheduler tasks and (for admin only) stop/resume the scheduler.
- 11: User panel. Click on user to edit profile.

	genome/eucar fasta,blast	Drosophila_mc	2011-05-17	2011-08-12 01:17:12	OK	1
	custom sql,fasta	gag_ensembl 63		-	-	2
	other xml	Gene_Ontolog	2011-08-16	2011-08-18 04:05:51	OK	3

Figure 2: Bank ownership color

- 1: Blue = Bank belongs to logged in user.
- 2: white = Bank does not belong to user nor to user group, but is still visible as it is public.
- 3: Purple = Bank does not belong to user but to a member of user group.

5.2. User management

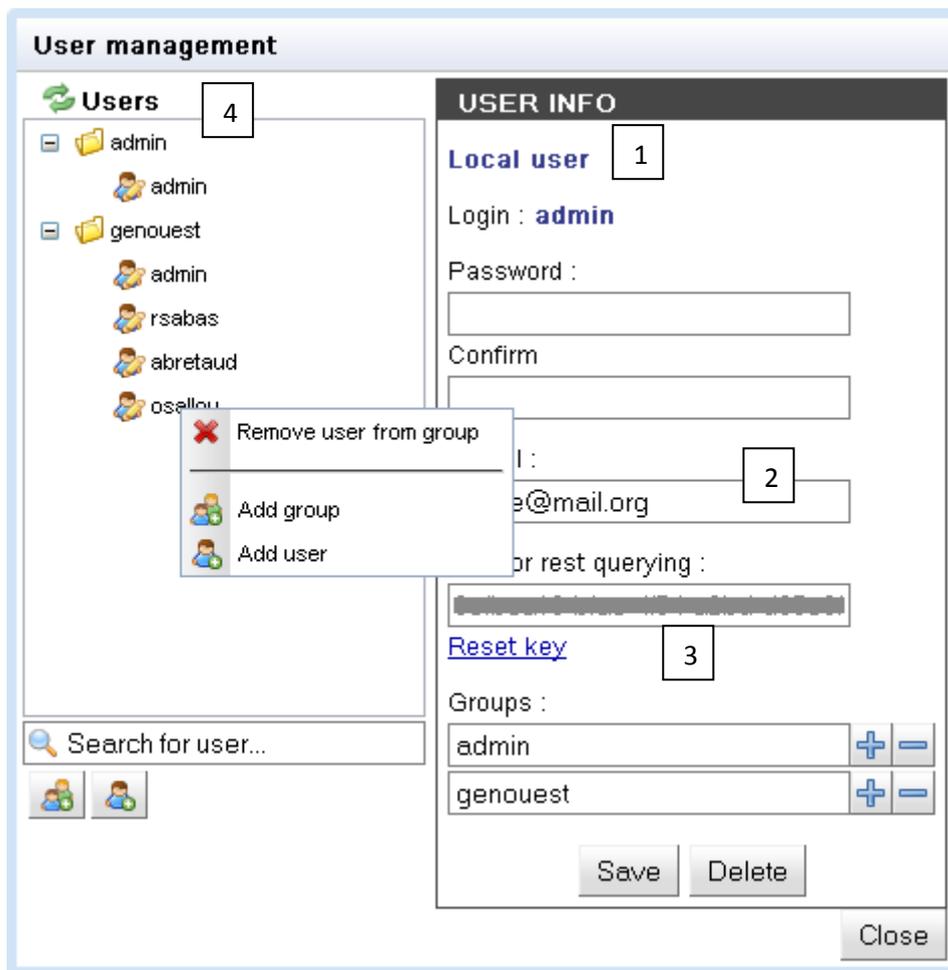


Figure 3: Admin panel

User has access to this panel when he clicks on his login. Presented above is the administrator panel that allows the administrator to manage users and groups.

1: User type. Only one authentication mode can be enabled:

- Local: Users are authenticated locally against the biomaj database. Admin has to create these users.
- LDAP: Users are authenticated with LDAP. The first time an LDAP user logs in, he is added to a group named after his login. Admin can later move him to the desired group.

When LDAP authentication is enabled, only the admin can be logged using local authentication.

2: User mail that is mainly used to notify user when his password has changed.

3: Authentication key that is used for REST services. That key can be reset at any time.

4: Users panel that shows groups and users. A user can belong to several groups.

5.3. Bank properties edition

5.3.1. Main panel

Bank editor

BANK SELECTION

You can choose to either create a new bank or edit an existing bank's properties :

Edit bank 1

Load from local :  2

Load from remote :  

Create new bank

SELECTED BANK : Acyrthosiphon_pisum-bcm

Options

See inherited properties 2

Override inherited properties

Include files : 3

Properties

Related files :  [Add a property...](#) 5

▼ Bank dependencies

	Depends bank	<input type="text" value="Drosophila_melanogaster-flybase"/>	6
	Files to move	<input type="text" value=".*"/>	
	Depends bank	<input type="text" value="IMG"/>	
	Files to move	<input type="text" value="flat/*.nin"/>	

Retrieve release from bank :

▶ Initialization 7

▶ Synchronization

▶ Pre-processes 

▶ Post-processes 

▶ Remove-processes 

8

Figure 4: BW bank edition

1: In this section you can:

- Edit banks properties loaded from :
 - The local repository
 - A remote repository which consists in the URL of a web page that biomaj will parse to retrieve any .properties file.
- Create a new bank from a predefined set of properties :
 - Mandatory: Mandatory only properties
 - Common: Properties commonly used
 - Exhaustive: All available properties

2:

- See inherited properties: show properties inherited from global.properties and any included properties file.
- Override inherited properties : allow inherited properties modification

3: The files located in `<workflows.dir>/<user login>/include` are listed and can be included to the currently loaded bank.

4: List all the selected bank related files: The bank file, the `global.properties` file and any other files that have been included.

5: List of the remaining properties that can be added to the bank.

6: Select what banks should be updated before the current one.

“Files to move” is a regular expression that specifies what files of the selected bank should be moved to the current bank production directory.

You can then specify if a specific bank release must be used as the current bank version (“retrieve release from bank”).

Detailed documentation is available in the Biomaj_1.2 user guide at section “4.4 Computed bank: bank dependencies”

7: This section contains the bank's properties organized by categories. Changing the related file changes the properties values.

8: The 'save' button allows you to:

- Override the loaded bank
- Save the bank under a new name
- Save the bank and run an update

5.3.2. Process edition panel

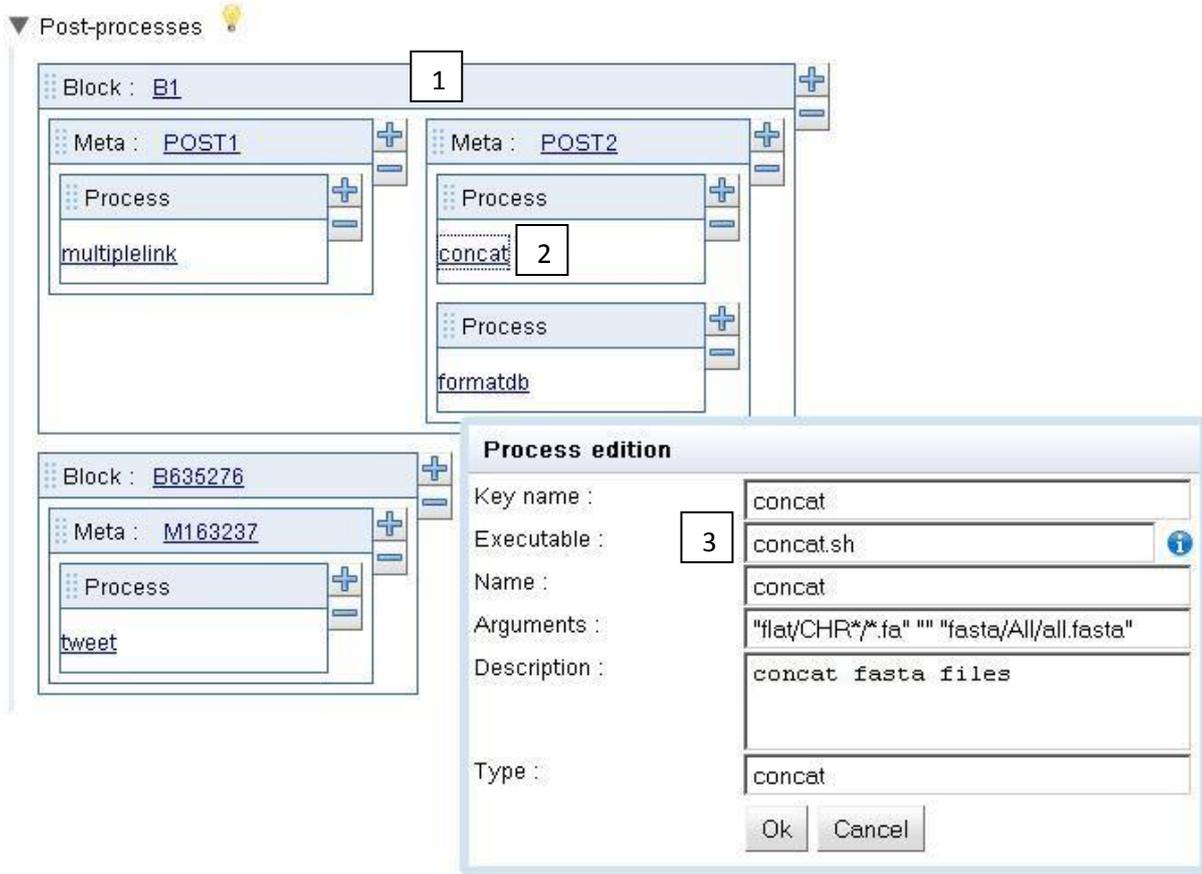


Figure 5: Process visualization

1: The processes are displayed in boxes with the following convention:

- From top to bottom is sequential (blocks and processes)
- From left to right is concurrent (meta-processes).

The boxes can be:

- reordered via drag&drop
- renamed : click on the link to change the name of a meta or a block
- Deleted : [-] button
- Added : [+] button

2: Clicking on the name of a process opens a dialog box that allows the user to edit the process properties.

3: The 'executable' field is a suggest box. It displays as suggestions any of the processes listed in the BioMAJ `process.dir`. Selecting a process that way will auto fill the name and description fields if a description was provided: process descriptions have to be entered manually in the file `processDescription.properties` located at BW root. It has the following structure:

```
process_name_1=process description.  
process_name_2=process description
```

...

5.3.3. Site manager

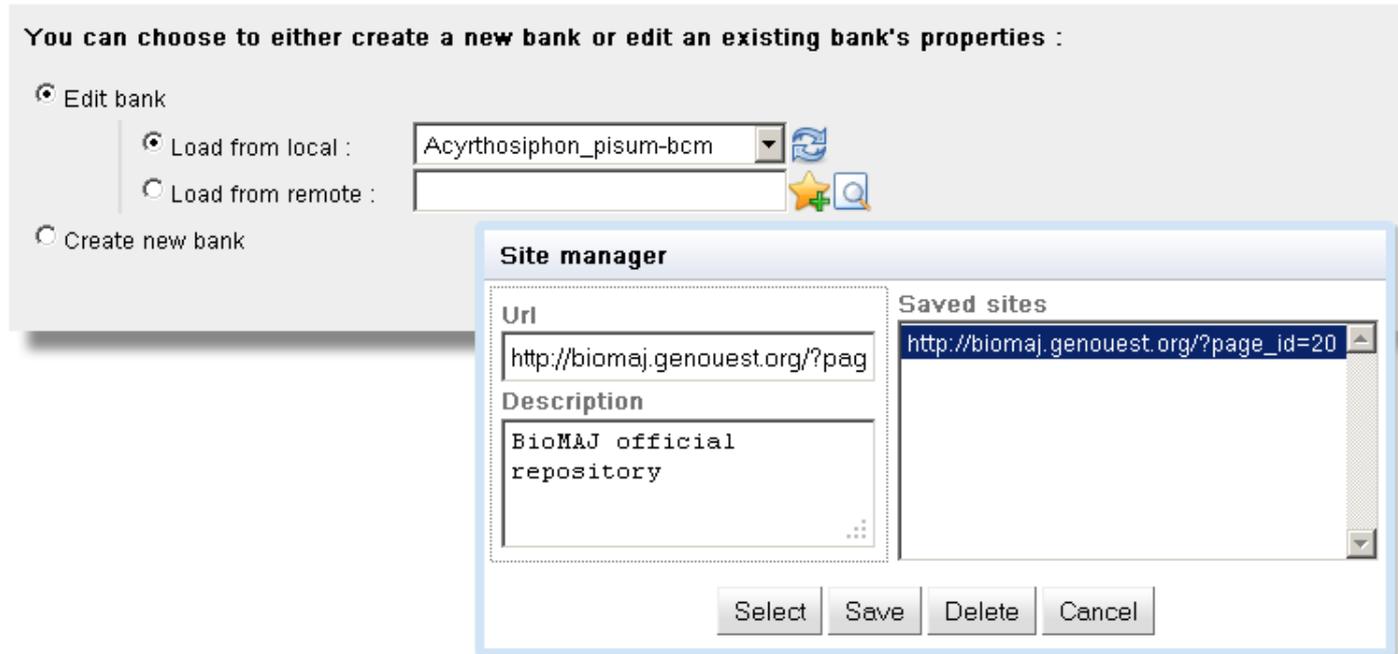


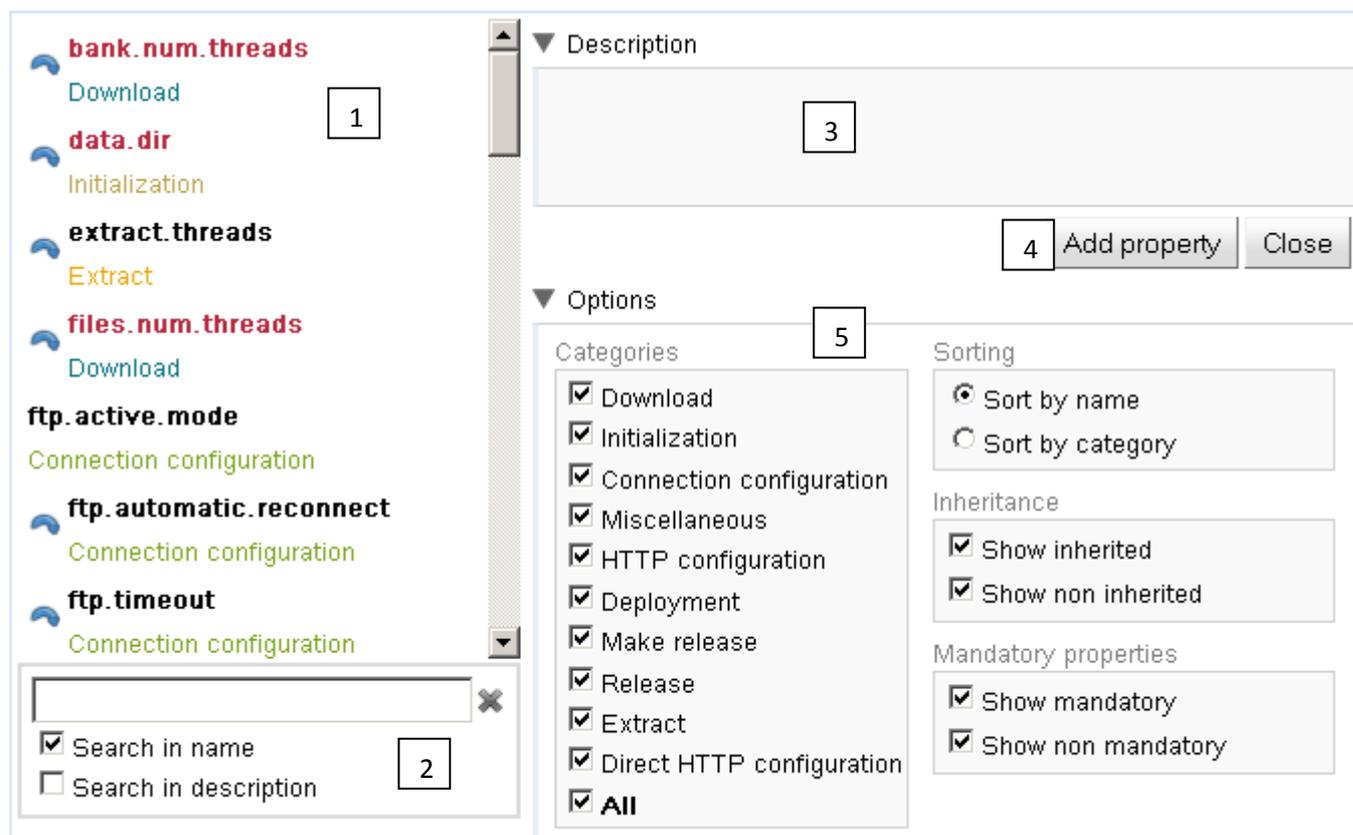
Figure 6: Remote repositories manager

The site manager panel shows up when the user clicks on the star.

In this panel the user can save remote repositories URL to easily load them later on.

To load the banks from such a repository, select the site in the list on the right, click the "select" button then click on the icon next to the star.

5.3.4. Properties selection panel



[Add a property...](#)

Figure 7: Properties selection panel

This panel contains all the properties available for the selected bank along with several options to filter these properties.

1: Properties list with the category they belong to:

- Properties in red are mandatory
- Each category has its own color
- The blue arrow next to a property indicates that it is already defined in a parent file (generally global.properties)

2: Search for properties by looking up their name and/or description.

3: Description for the selected property

4: Add the selected property to the bank or close the panel

5: Options to filter properties depending on:

- Categories
- Inheritance
- Mandatoriness
- Sort type : sort by property name or by category name

6. Running bank updates

6.1. Manual updates

You can run at any time an update for any selected bank in the status grid. The update options are available in the toolbar or in the contextual menu for a bank. You can:

- Start a normal update: -- update bank
- Start a new update (ignores previous unclosed sessions): --update bank --new
- Start an update from scratch: --update bank --fromscratch
- Rebuild a bank (run only the post-processes if they changed): --rebuild bank

6.2. Scheduled updates

Scheduled updates are run automatically at a given time defined by the scheduler in the web interface. For each task, a normal update is run for the complete bank list.

The syntax used to schedule jobs is very similar to the cron syntax. The main difference is that it can handle seconds while cron smallest time unit is the minute.

<http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html>

7. REST Querying service

The following documentation is also available at <BW_URL>/rest.html

7.1. Retrieving bank information

BW provides a REST service that allows the user to retrieve information on the banks in JSON format via the URL. You can get 3 types of result:

7.1.1. The formats list

Syntax: <BW URL>/GET?formats

```

{
  "format":[
    {
      "value":"blast"
    },
    {
      "value":"emboss"
    },
    {
      "value":"fasta"
    }
  ]
}

```

7.1.2. The types list

Syntax: <BW URL>/GET?types

```

{
  "type":[
    {
      "value":"aphidbase/nucleic"
    },
    {
      "value":"aphidbase/proteic"
    },
    {
      "value":"gene/duplicate"
    }
  ]
}

```

7.1.3. The banks list

7.1.3.1. General structure

The structure of the returned JSON object is the following:

```

{
  "banks":[
    {
      "name":"bank name",
      "session_date":"session date",
      "current_release":"release number",
      "releases":{
        "release number":{
          "path":"directory path",
          "formats":[
            {
              "value":"format 1",
              "sections":[
                {
                  "name":"section name",
                  "files":[
                    "filepath_1",...,"filepath_n"
                  ],
                  "sections":[
                    ...
                  ]
                },...
              ]
            }
          ],...
        }
      ]
    },
  ]
}

```

```

        {
            "value":"format 2",
            "sections":[
                ...
            ]
        }
    ],
    ...
},
"db_type":"bank type"
}...
]
}

```

7.1.3.2. Section element

In further details, for each format, a list of the associated files is returned as a section element. A section structures the files according to their directory structure.

Example:

Directory structure:

```

fasta
|- dir_1
|  |- dir_11
|  |  |- file_a
|  |  |- file_b
|  |- dir_12
|  |- file_c
|- dir_2
|- file_d
|- dir_21
|- file_e

```

Sections :

```

"formats":[
  {
    "value":"fasta",
    "sections":[
      {
        "name":"dir_1",
        "sections":[
          {
            "name":"dir_11",
            "files":[
              "file_a","file_b"
            ]
          },
          {
            "name":"dir_12",
            "files":[
              "file_c"
            ]
          }
        ]
      },
      {
        "name":"dir_2",
        "files":[
          "file_d"
        ]
      }
    ]
  },
  ...
]

```

```

    "sections": [
      {
        "name": "dir_21",
        "files": [
          "file_e"
        ]
      }
    ]
  }
]

```

If you don't want the the sections to be shown you can use the **lightmode** parameter: only the formats name will be returned.

```

"formats": [
  {
    "value": "fasta"
  },
  {
    "value": "blast"
  }
]

```

7.1.3.3. Filtering the results

You can specify 4 parameters to retrieve the bank list:

- **banks** (mandatory) : Can be either the bank name or "all" to retrieve all the banks.
- **formats** (optional) : Can be either a specific format or "all". Multiple formats are separated with | character.
- **types** (optional) : Can be either a specific type or "all". Multiple types are separated with | character.
- **lightmode** (optional) : Use these parameter if you don't need the formats directories to be listed.
- **nocache** (optional) : Do not use cache. This will cause the cache to be refreshed. The cache is implemented as files named "listing.<format>" located in each bank production directory. Deleting this files will also force the "nocache".
- **key** (optional) : Authenticate with an user key. If not provided, only the public banks are available.

Examples:

- All banks with fasta format:
<BW URL>/GET?banks=all&formats=fasta
- All banks with fasta format and nucleic type:
<BW URL>/GET?banks=all&formats=fasta&types=nucleic&lightmode
- All banks with fasta or xml format and nucleic or proteic type :
<BW URL>/GET?banks=all&formats=fasta|xml&types=nucleic|proteic
- Genbank bank only if it has fasta format:
<BW URL>/GET?banks=genbank&formats=fasta
- Genbank with no cache :
<BW URL>/GET?banks=genbank&nocache

7.2. Downloading files

The REST service can also be used to download bank files or directory.

The URL syntax is:

```

<BW URL>/GET?bank=<bank_name>&path=<path>[&version=<version>&zip=<true|false>&key=<user_key>]

```

- **path** and **bank** parameters are mandatory.

- **bank** is the bank name to consider.
- **version** is the version of the bank to consider. If not specified, default value is current
- **path** is the relative path to the file or directory to download (relative to <bank>/<version>, e.g. flat/dir/file1). To download the whole version, leave that parameter empty.
- If **path** points at a directory, a zip archive will be provided by default.
- **zip** parameter only applies to files.
- If the bank is private, an authentication key is required. This is done with **key** parameter. Access will be granted only if you belong to the group the bank owner belongs to. Your key is available in your user panel in the web interface.

7.3.Starting updates

Standard bank updates can be run via the REST service if you are authenticated and have access to the bank:

<BW URL>/GET?update=<bank_name>&key=<user_key>