



# PatMOB\_06 TUTORIAL

## Table of Contents

<b>1. Introduction</b>	<b>2</b>
1.1. What is PatMOB?	2
1.2. System requirements	2
1.3. Installation	2
1.4. Launch	2
<b>2. Plug-ins</b>	<b>4</b>
2.1. PatMOB plug-in architecture	4
2.2. REST@OPS plug-in	4
2.3. OPS Web Services	5
<b>3. Tree branch editor</b>	<b>6</b>
3.1. Overview	6
3.2. Patent info display	6
3.3. Popup menu	6
3.4. Drag and drop	7
3.5. Menu bar	7
<b>4. Main application window</b>	<b>8</b>
4.1. Overview	8
4.2. File menu	8
4.3. Help menu	9
4.4. Plugins menu	9
<b>5. Trouble-shooting</b>	<b>10</b>
5.1. Proxies	10

## 1. Introduction

### 1.1. What is PatMOB?

PatMOB is a software platform enabling rapid development of software modules (plug-ins) designed for management of various patent data formats. It combines an intuitive graphical user interface (GUI), with an application programming interface (API) exposing a rich core infrastructure for plug-in development. The use of custom designed PatMOB plug-ins for patent landscaping has been recently described (Masiakowski P, Wang S, Integration of software tools in patent analysis, World Patent Information (2013), <http://dx.doi.org/10.1016/j.wpi.2012.12.010>, in press). Also, see <http://genemob.com/patent/> for additional information and new developments.

PatMOB\_06 is deployed as a desktop application. Version 0.6 designation is meant to imply that it is a little more than half-way to the stable first release, 1.0.

### 1.2. System Requirements

Written in the Java programming language, PatMOB can be run on Windows, Mac OS or Linux. PatMOB\_06 is built with the most recent release of Java, version 7 (also referred to as 1.7; for details, see <http://www.oracle.com/technetwork/java/javase/downloads/index.html>). However, no new language features introduced in Java 7 are used in PatMOB code, and it has been extensively tested on the still popular Java version 6 (1.6). PatMOB\_06 has not been tested, and it may not run properly, on Java version 5 or earlier.

This tutorial has been developed on Windows 7, but it will work with only slight modifications on Linux or Mac OS.

### 1.3. Installation

PatMOB\_06 is distributed as a ZIP archive. Extract the enclosed *patmob\_06* folder to any directory on your system. This folder contains the *PatMOB\_06.jar* file and the *lib* folder. All the Java classes comprising the PatMOB\_06 program are in the *PatMOB\_06.jar* archive. The *lib* folder contains various Java libraries required to run PatMOB. It also contains the *OPS\_ClientPlugin.jar* archive with the preinstalled REST@OPS plug-in described below.

### 1.4. Launch

On Windows and Mac OS, PatMOB can be run simply by double-clicking the *PatMOB\_06.jar* file. An alternative way, especially in case of any problems, is to run it from the command line. Below is an example session in Windows Command Prompt (use Terminal on Linux or Mac), in which I change directory to the *patmob\_06* folder on a removable disk (D:) and launch the program:

```
C:\Users\Piotr>D:
D:\>cd patmob_06
D:\patmob_06>java -jar PatMOB_06.jar
Created tables in Derby PatmobDB
Found plugin REST@OPS plugin
```

Started that way, PatMOB will print diagnostic messages. In the example above, it reported a successful creation of the database and installation of a plug-in.

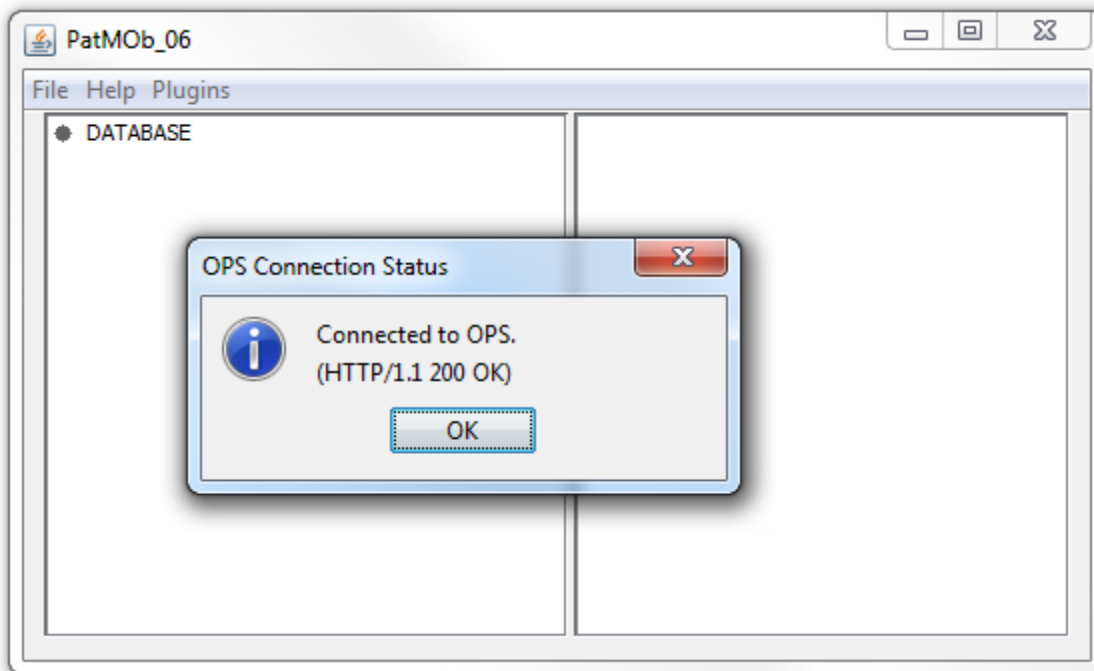
On its first run, PatMOB\_06 will create a *patmob\_06\_data* folder in the default user directory (e.g. *My Documents* on Windows). *patmob\_06\_data* folder contains the *PatmobDB* folder, with the installation of the relational database, Java DB, which is a part of the standard Java distribution from Oracle. The *derby.log* text file has the details of the installation, and it is so named since Java DB is being developed as an open source project Apache Derby by the Apache Software Foundation (<http://db.apache.org/derby/>). *patmob\_06\_data* folder also contains another text file, *patmob.properties*. The program uses this file to store configuration data. Users can open this file with a word processor and modify configuration details.

Thus, the final working installation of PatMOB\_06 on your system will consist of 2 directories:

```
patmob_06 folder
  PatMOB_06.jar program file
  lib folder with libraries

patmob_06_data folder
  PatmobDB folder with Java DB (Derby) files
  derby.log text file
  patmob.properties text file
```

At each launch, PatMOB\_06 tries to connect to the Open Paten Services (OPS) RESTful Web Services at the European Patent Office (EPO) to enable real-time patent information lookup. If there is an Internet connection and everything goes well, you should see something like the following screenshot:

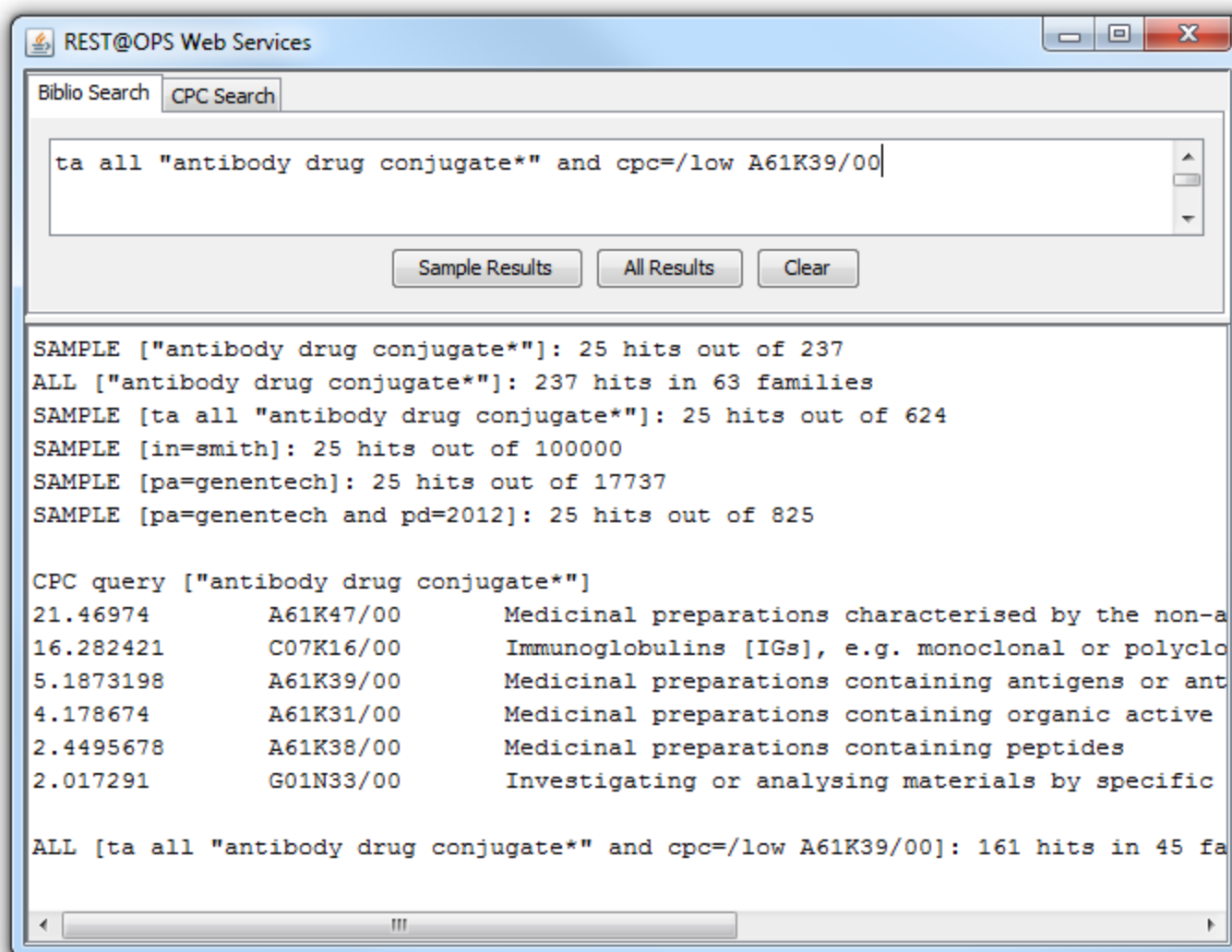


Click OK, and proceed to the next chapter to explore the REST@OPS plug-in.

## 2. Plug-ins

### 2.1. PatMOB plug-in architecture

PatMOB plug-ins are Java programs -- ranging from small snippets written ad hoc to address the emerging needs, to full programs with their own business logic and complex GUI -- running in the PatMOB framework. The included REST@OPS plugin demonstrates the seamless integration of OPS Web Services at EPO with the growing functionality of PatMOB core infrastructure through the PatMOB API.



### 2.2. REST@OPS plug-in

Open the Plugins menu and select the *REST@OPS plugin*. Enter a common query language (CQL) query in the top text area of the *Biblio Search* tab, or some keywords in the *CPC Search* tab. CQL is quite intuitive for a patent information professional even without reading the manual, as shown in the lower text area of the screenshot below. Note that the *Sample Results* button retrieves first 25 publications for the query and the *All Results* button -- all publications, organized into simple patent families. The results retrieved from OPS are displayed in the tree branch editor which is described in the next chapter.

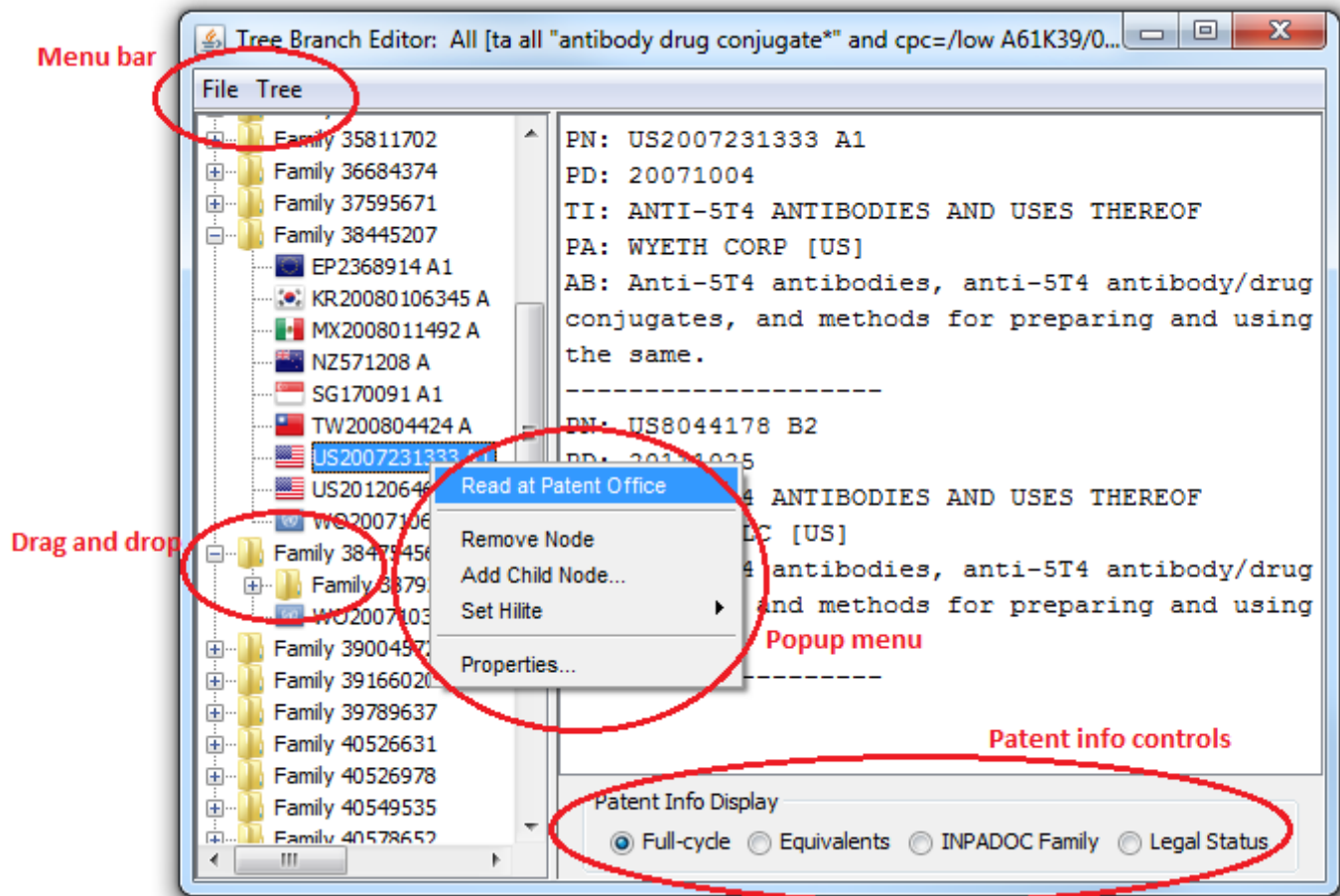
### 2.3. OPS Web Services

OPS is an excellent, free of charge Internet resource from the European Patent Office, offering real-time programmatic access to the production stable EPO patent data; version 3.0 introduces support for the Cooperative Patent Classification (CPC). OPS implements all services with an elegantly designed and well documented and maintained REST architecture, described at <http://www.epo.org/searching/free/ops.html>. In particular, chapter 4.2. *CQL index catalogue* in the OPS reference guide available at the OPS website ([http://documents.epo.org/projects/babylon/eponot.nsf/0/2F88B7285FC1E3ECC125785500531278/\\$File/OPS\\_v3\\_0\\_documentation\\_v1\\_1\\_en.pdf](http://documents.epo.org/projects/babylon/eponot.nsf/0/2F88B7285FC1E3ECC125785500531278/$File/OPS_v3_0_documentation_v1_1_en.pdf)) describes the use of the common query language (CQL) in published-data service bibliographic search. CQL is used to retrieve patent information in the REST@OPS plug-in below.

### 3. Tree branch editor

#### 3.1. Overview

The tree branch editor is a standard component of PatMOB infrastructure. PatMOB arranges all patent data in trees and the editor gives the user full control over the tree structure and content. In particular, the editor allows extensive review and curation of massive amounts of patent data loaded from OPS and a wide variety of other sources. The main features of the tool are outlined in the screenshot below, displaying the results of the last query in the previous chapter.



#### 3.2. Patent info display

Clicking on a patent publication node in the tree automatically fetches the real-time information from OPS. The default is the publication full-cycle summary shown above, but the radio buttons in the *Patent Info Display* panel allow access to patent equivalents, INPADOC family and legal status information.

#### 3.3. Popup menu

Right-click on a tree node displays a contextual popup menu with the following items:

- **Read at Patent Office.** This item appears when the tree node corresponds to a patent publication. Selecting it opens, in the default system browser, a web page for that publication - at full-text USPTO web site for the US documents, and EPO Espacenet for the others.

- **Remove Node.** Deletes the node from the tree.
- **Add Child Node.** Opens the node properties dialog (see chapter 4) allowing creation of a new node of a desired type.
- **Set Hilite.** Allows font color selection for the node label.
- **Properties.** Opens the properties dialog (see chapter 4) to rename node and add notes or documents.

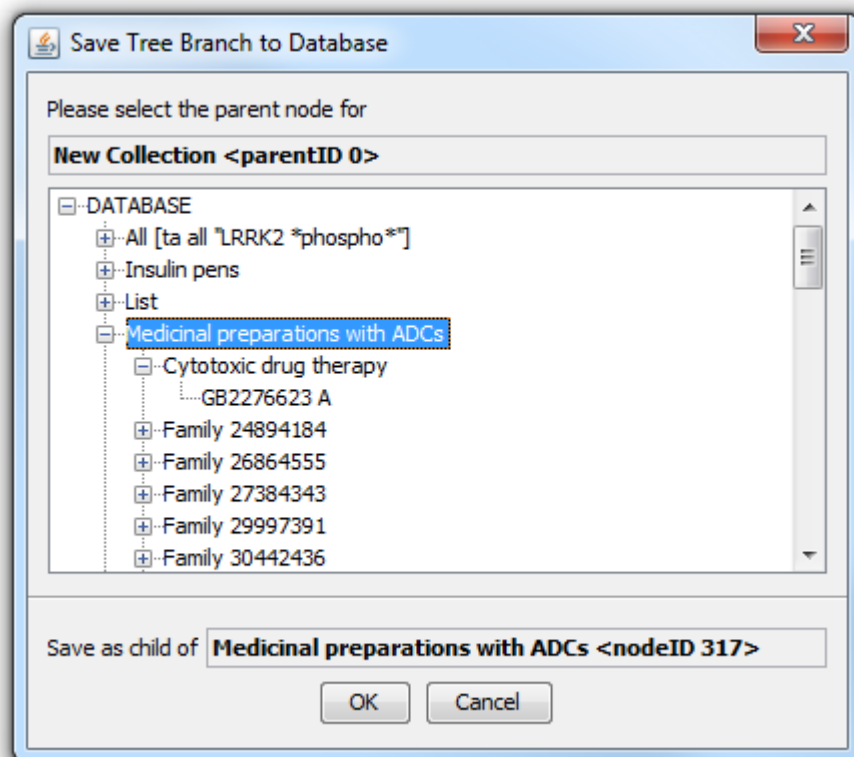
### 3.4. Drag and drop

A tree node, and all its children, can be moved to another location in the tree with the familiar gesture of the mouse. In addition, tree nodes can be dragged between different editor windows.

### 3.5. Menu bar

The items selected from the File and Tree menus perform the following actions on the whole tree:

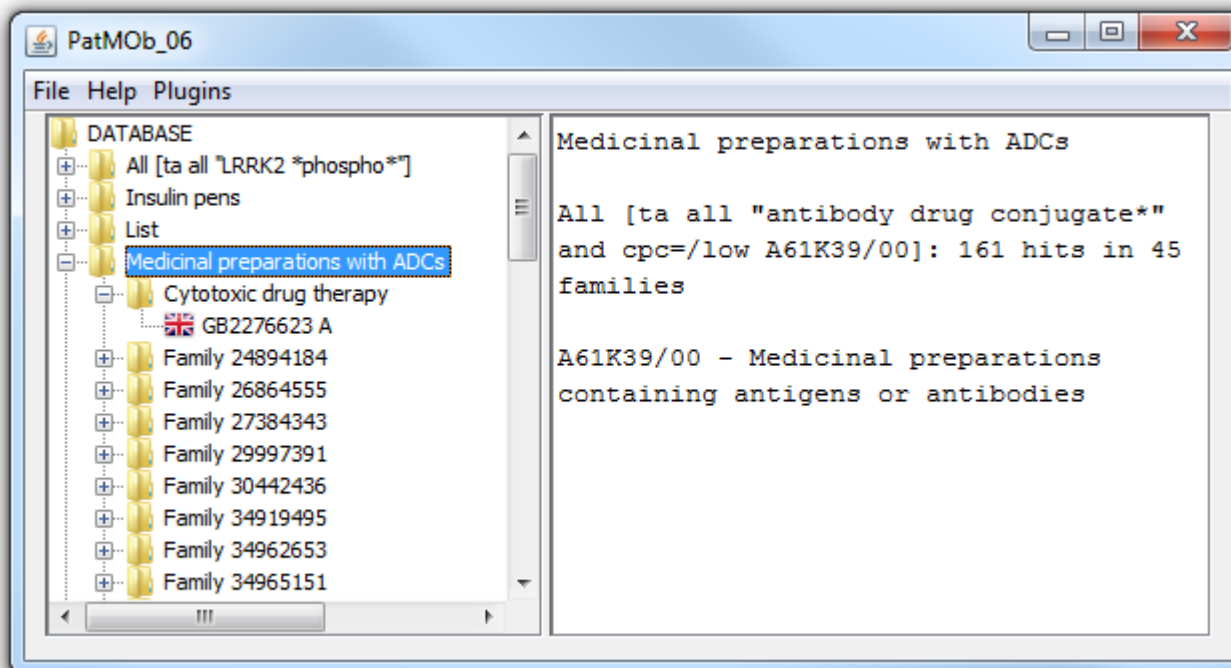
- **Tree Menu: Convert to FreeMind Map...** Saves the tree as a mind map for the free mind-mapping software FreeMind ([http://freemind.sourceforge.net/wiki/index.php/Main\\_Page](http://freemind.sourceforge.net/wiki/index.php/Main_Page)).
- **File Menu: Write to Text File...** Saves the tree data in a simple text file.
- **File Menu: Write to XML File...** Saves the tree data in a native PatMOB XML format.
- **File Menu: Save to Database...** Saves the tree data to the Java DB database. Opens the dialog allowing insertion at any point in the entire database tree:



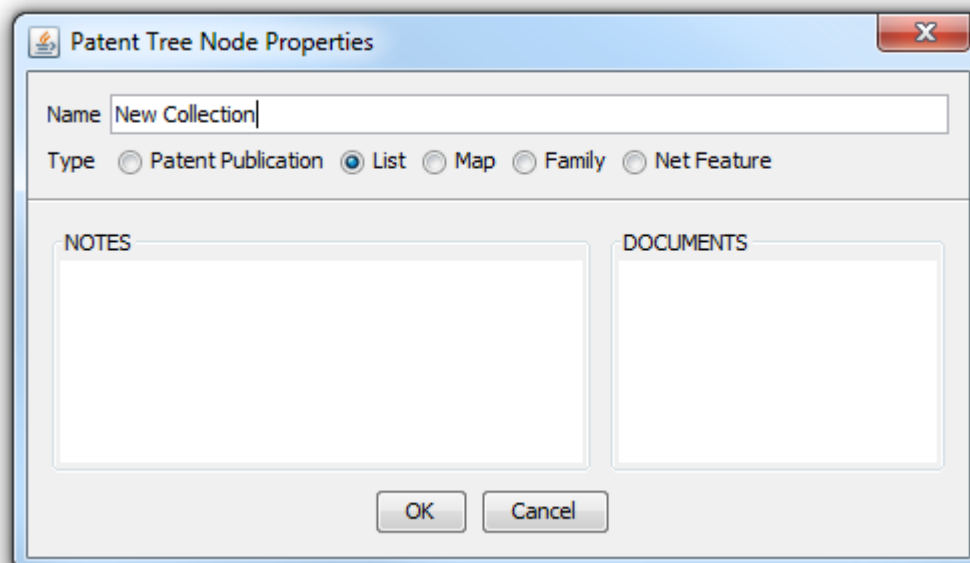
## 4. Main application window

### 4.1. Overview

The main window displays the entire database tree in its left-side panel. Clicking on a patent node displays the publication full-cycle info fetched in real-time from OPS and not saved in the database. For other types of nodes, the content of the database NOTES field is displayed. That field holds up to 500 characters of text.



### 4.2. File menu



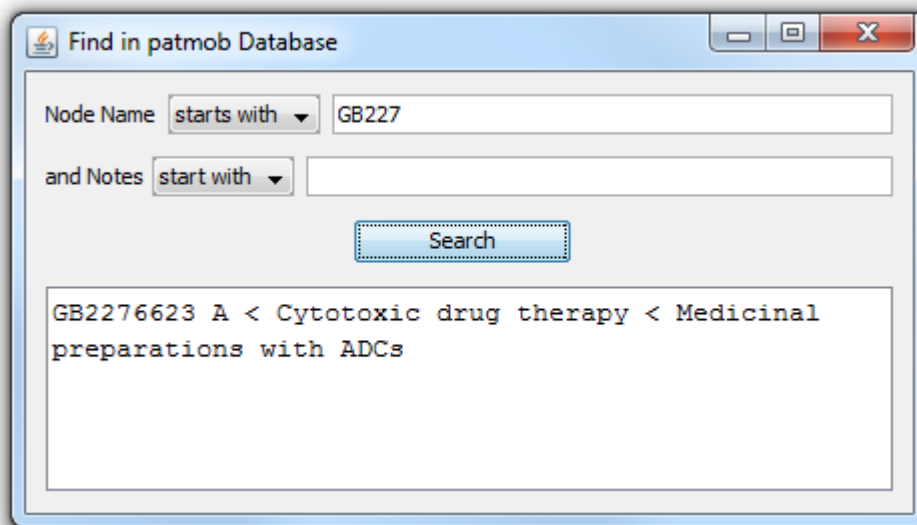


The following items are available on the File menu:

- **New Collection...** Opens a new tree branch editor window with a newly created tree node as a root. The user determines the node type by selecting the appropriate radio button in the properties dialog above.
- **Load from XML...** Allows user to open a file in the native PatMOB XML format in a new tree branch editor.
- **Edit Node...** Opens the selected tree branch in the editor window.
- **Delete Node...** Deletes the selected node and all its children from the database.
- **Exit** Quit the program.

#### 4.3. Help menu

- **Find in DB...** Opens a window for searching the internal database:



- **About...** Displays PatMOB version information.

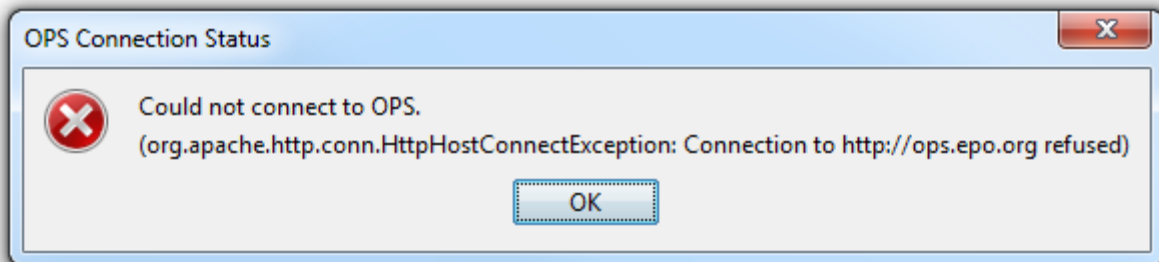
#### 4.4. Plugins menu

All the available plug-ins are accessed from this menu. Plug-ins can be included in PatMOB distribution, like the REST@OPS plug-in, or installed by specifying their names in the patmob.properties file. In the latter case, a JAR file with the plug-in classes has to be located in the program's classpath.

## 5. Trouble-shooting

### 5.1. Proxies

Various institutions protect their computer networks with proxy servers or firewalls. If you run PatMOB on a Local Area Network with a proxy server, you may get the following error message when trying to access OPS:



PatMOB\_06 supports the Microsoft NTLM authentication scheme. Ask your IS support for assistance, in particular the address of the proxy, which will look like "myproxy.mycompany.com:1234". Enter this address on a separate line in the patmob.properties file in exactly the following format:

```
patmobProxy=myproxy.mycompany.com\ :1234
```

When asked for authentication by the proxy, PatMOB will display a login dialog. Enter the user name, password and domain name you use to access your network.