Proline Installation Guide

Release 1.7-2.0

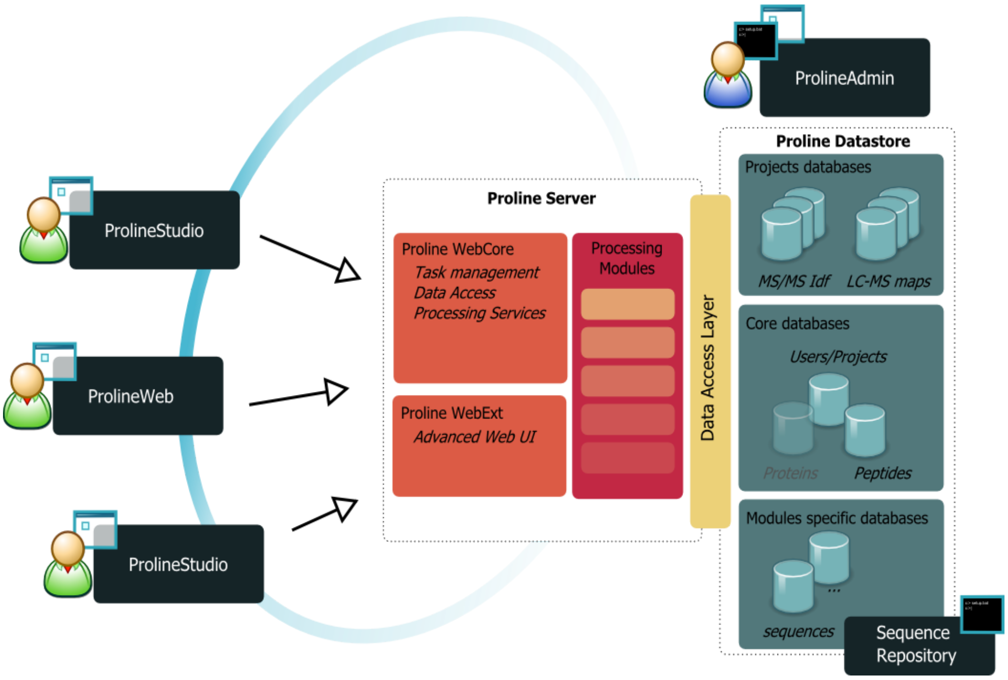
Table of content

Introduction

Architecture Overview

The suite is based on different components (see figure below):

- **Proline Datastore** is based on a Relational Database Management System storing the data used by the software. Proline stores data in three different database schemas. The "core" database schemas created once at datastore initialization. It contains data related to users' projects (**UDSdb**). The two additional schemas are used to create a new database each time a new user project is created. Those databases store identification data, peptides sequences, post-translational modifications (**MSIdb**) and quantification data (**LCMSdb**) associated with users projects.
- **Proline Server** is the core component handling user's processing tasks. The server is associated with a messaging system responsible for sending and receiving of messages between the graphical user interfaces and the server.
- **ProlineStudio** and **ProlineWeb** are two different graphical user interfaces, both allowing users to start tasks and visualize the resulting data. Either or both can be used:
    - **ProlineStudio** is a desktop application that must be installed on each individual user's computer.
    - **ProlineWeb** is a web application requiring an additional web server installation, but can be access by a standard web browser on the client side.
- Proline **Sequence Repository** is an external module to retrieve protein sequences from Fasta files and store them in a sequence database.
- **Proline Admin** is a system administration application to setup and manage the Proline suite. This application is available as a command line application or with a graphical user interface.



Requirements

The server-centric architecture of Proline imposes different requirements for the server computer and the client computers.

- Server-side Proline requirements:
  - A Java SE 8 JRE must be installed.
  - The PostgreSQL database server (tested versions are 9.4) must be installed and configured. On Windows the installer includes the PostgreSQL server, which can be installed on the same computer than Proline or a distinct one. By default, PostgreSQL settings are defined for modest hardware configurations but they can be modified to target more efficient hardware configurations (See PostgreSQL optimization and authorizations).
  - Proline Server **must run in English "locale"**, on Linux OS, environment variable `LANG=en_US.UTF-8` can be exported before starting Proline Server or the `start_cortex` script can be modified to add `-Duser.language="en" -Duser.country="US"` parameters.
- Client-side requirements for Proline Studio:
  - A Java SE 8 Runtime Environment (JRE) must be installed (**on Windows, Proline-Studio installer already includes a JRE 32 or 64-bit distribution**)

If you want to use Proline remotely through the Web client, the ProlineWeb components and their requirements must also must installed.

- Server-side requirements for Proline Web :
  - MongoDB database server must be installed (at least version 2.6.x). Note: this database server can be installed on a distinct computer.
- Client-side requirements for Proline Web:
  - A recent Web Browser (IE 9+, Firefox 25+, Chrome 30+).

Installing Proline Server

The Proline Server installation process can be done using the Windows installer or manually using archive files.

Once installation have been done, setup should be done. You can either use the graphical interface Proline Admin GUI or use the command-line tool Proline Admin to set up Proline.

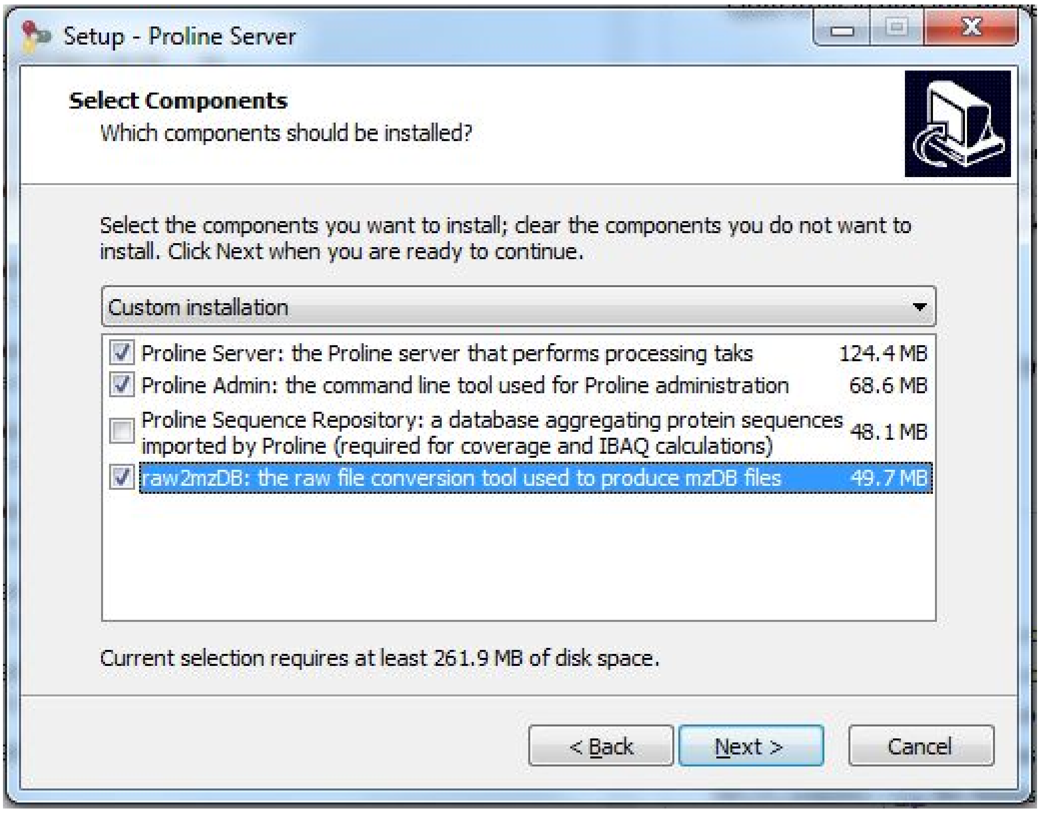
Installing Proline Server using the Windows installer

Download the automated installer from the Proline website (http://proline.profiproteomics.fr/download/).

Warning:two different installers are provided, one with Postgresql server and one without. To install Postgresql server you should have sufficient rights and you should run the executable file as "administrator". If you don't have sufficient rights or if Postgresql is already installed choose the installer without postgresql.

The wizard will guide you through the installation process. By default, the installer will unpack all components on the computer. However, it is possible to install the Proline components on distinct computers if it fits better your hardware architecture.

It is recommended to put Proline Server and Proline Admin on the same computer. Sequence Repository is recommended to be installed on the computer where Fasta files are accessible.



Setup - Proline Server

Select Components

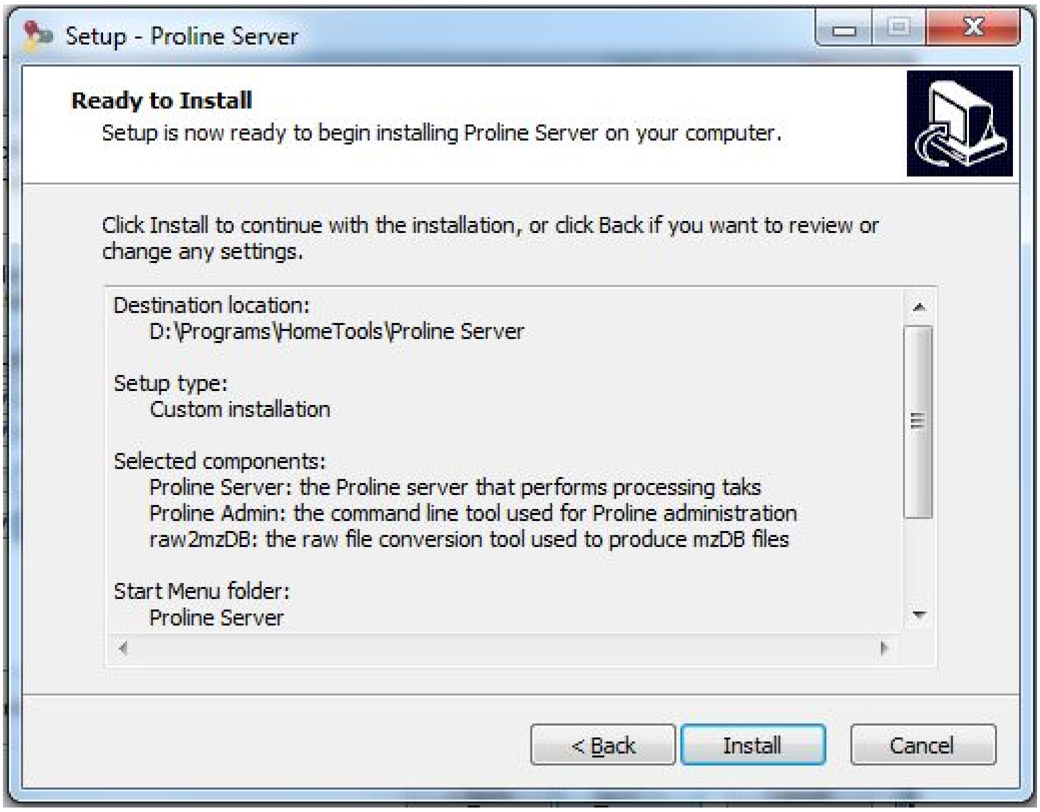
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom installation

☑	Proline Server: the Proline server that performs processing taks	124.4 MB
☑	Proline Admin: the command line tool used for Proline administration	68.6 MB
□	Proline Sequence Repository: a database aggregating protein sequences imported by Proline (required for coverage and IBAQ calculations)	48.1 MB
☑	raw2mzDB: the raw file conversion tool used to produce mzDB files	49.7 MB

Current selection requires at least 261.9 MB of disk space.



Ready to Install

Setup is now ready to begin installing Proline Server on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

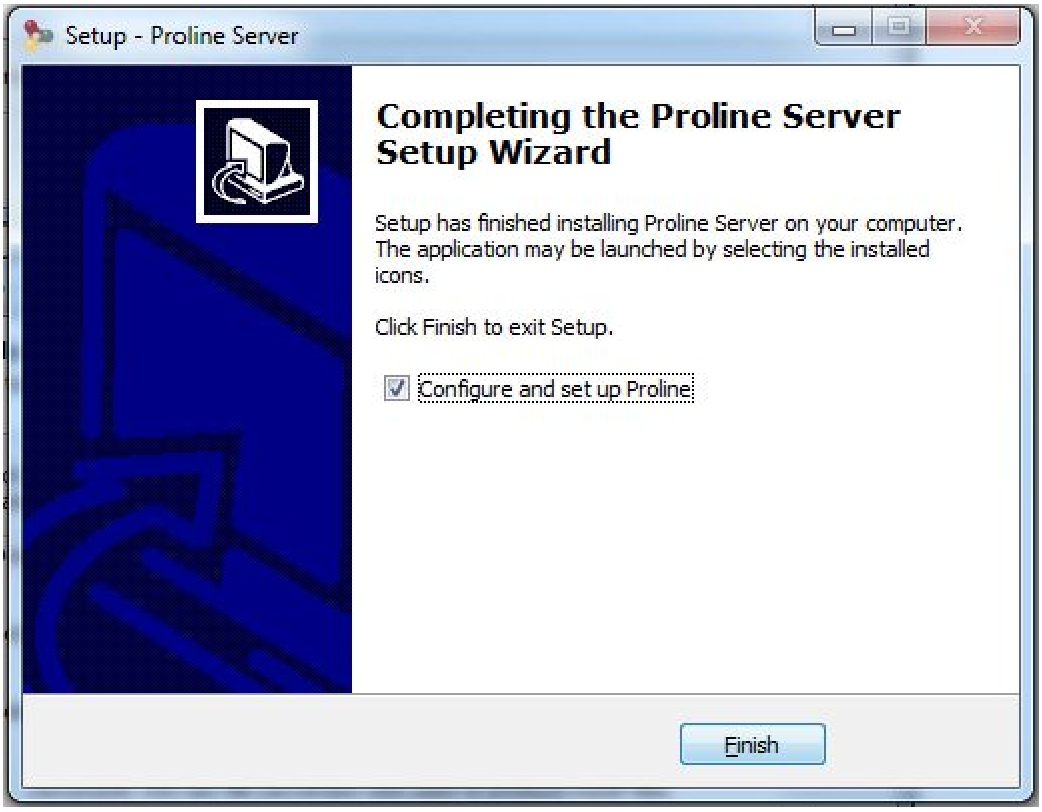
Destination location:
    D:\Programs\HomeTools\Proline Server

Setup type:
    Custom installation

Selected components:
    Proline Server: the Proline server that performs processing taks
    Proline Admin: the command line tool used for Proline administration
    raw2mzDB: the raw file conversion tool used to produce mzDB files

Start Menu folder:
    Proline Server

When components are installed, the installer propose you to open the configuration tool (Proline Admin GUI). After a first install, Proline Server requires configuration and Proline Datastore needs to be setup.



Completing the Proline Server Setup Wizard

Setup has finished installing Proline Server on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Configure and set up Proline

Warning: Before running setup you should **create a specific Postgresql user** for Proline with "create database" rights. At installation, login/password for postgresql user is postgres/postgres.
You can keep your installer windows open, create the Postgresql user using pgAdmin 3 (<pg_install_path>\PostgreSQL\9.4\bin\pgAdmin3.exe) or using createuser.exe in a command Window(<pg_install_path>\PostgreSQL\9.4\bin\createuser.exe)

When clicking on Finish, The Proline-Admin GUI install will be automatically started, in order to help you to configure your installation.

Proline-Admin GUI

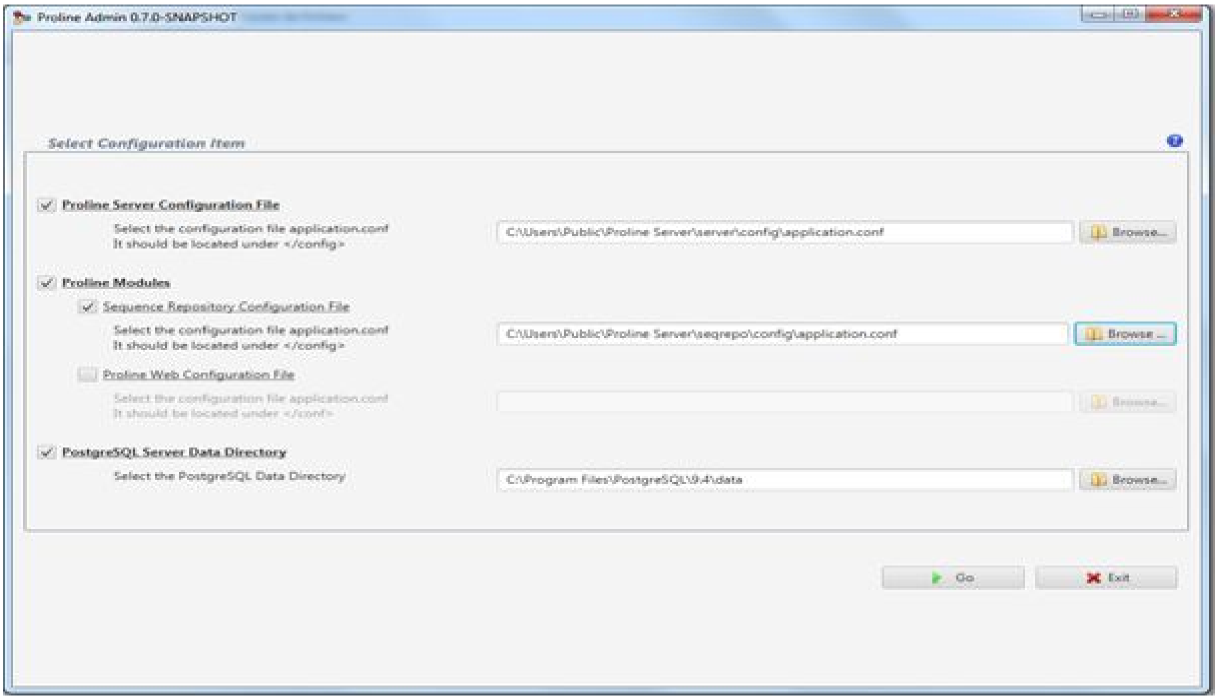
Proline-Admin GUI is a graphical tool that allows Proline administrator to configure each component. It includes the edition of the configurations files, the connection to the database and management of PostgreSQL via Proline-Admin GUI install. It allows Proline administrator to monitor and to manage Proline databases via Proline-Admin GUI monitor as well.

Launch Proline-Admin GUI install

Proline-Admin-GUI install should be automatically executed after installing Proline. If you have to use it afterwards, execute the script `install.sh` (if you are using Linux) or `install.bat` (if you are using Windows) located in the <proline_install>/Proline-Admin folder.

Select configuration component

In this first step you need to select which component of Proline you want to configure. For a standard installation of Proline on a single server, you should configure all of them but you can also choose to use a specific PostgreSQL server for instance.



Proline Admin 0.7.0-SNAPSHOT

Select Configuration Item

☑ Proline Server Configuration File

Select the configuration file application.conf
It should be located under </config>

C:\Users\Public\Proline Server\server\config\application.conf

☑ Proline Modules

☑ Sequence Repository Configuration File

Select the configuration file application.conf
It should be located under </config>

C:\Users\Public\Proline Server\seqrepo\config\application.conf

□ Proline Web Configuration File

Select the configuration file application.conf
It should be located under </conf>

☑ PostgreSQL Server Data Directory

Select the PostgreSQL Data Directory

C:\Program Files\PostgreSQL\9.4\data

"Proline Server Configuration File": The full path to Proline server configuration file.
Its default location is <proline_install>/server/config/application.conf

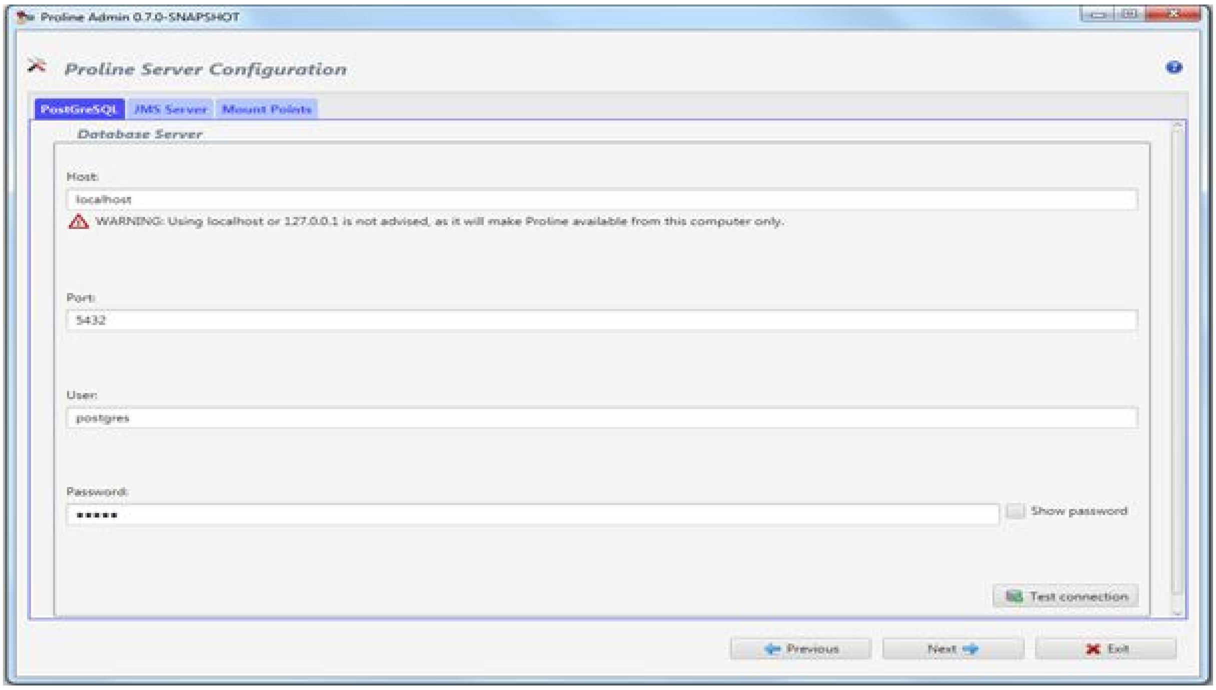
Proline Modules:

- “Sequence Repository Configuration File”: the full path to the Sequence Repository Configuration file its default location is <proline_install>/seqrepos/config/application.conf
- “Proline Web Configuration File”: the full path to the Proline web configuration file its default location is <proline_install>/web/config/application.conf

“PostgreSQL Server Data Directory”: the full path to the PostgreSQL data directory. This is the directory containing the configuration files `postgresql.conf` and `pg_hba.conf`.

Proline Server Configuration

PostgreSQL



Host:
localhost
WARNING: Using localhost or 127.0.0.1 is not advised, as it will make Proline available from this computer only.

Port:
5432

User:
postgres

Password:
•••••

□ Show password

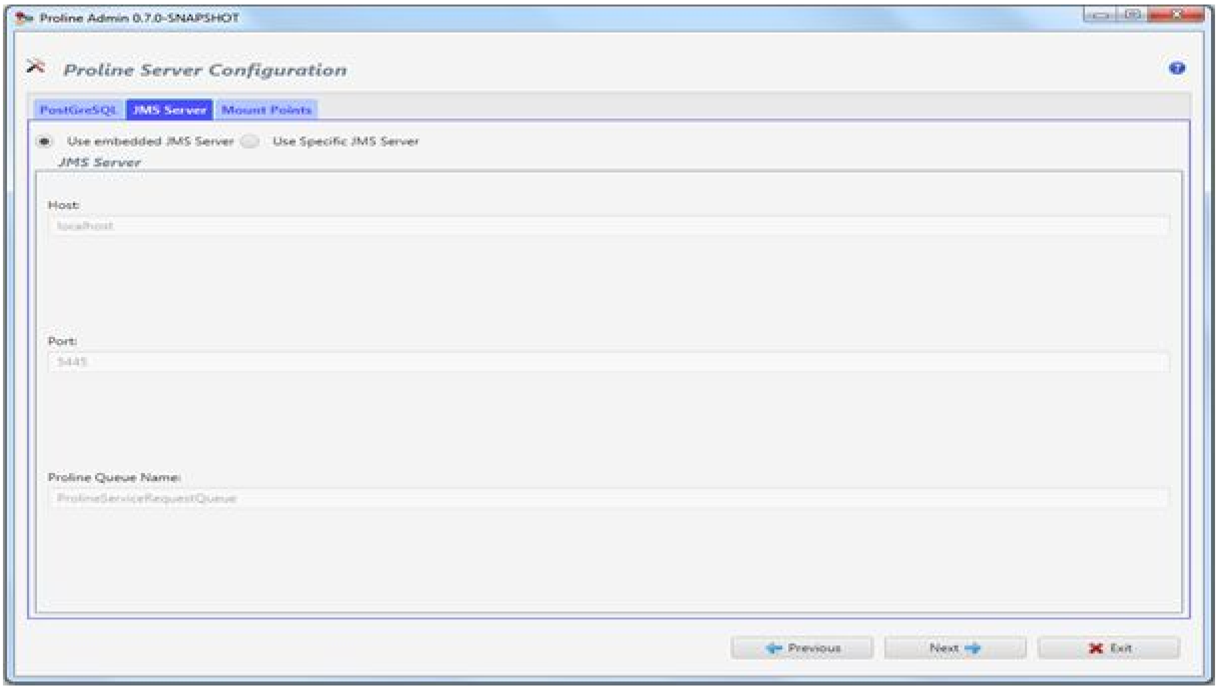
The PostgreSQL tab lets you edit the parameters to connect to the Proline databases:

- “Host”: the host name of the server running PostgreSQL. Using localhost or 127.0.0.1 is strongly discouraged, Proline databases will be available for this computer only.
- “Port”: the port number 5432 is the default value.
- “User”: the PostgreSQL user name to connect to the database.
- “Password”: the user password to connect to Proline database. You can use the button “Test connection” and Proline-Admin will try to connect to PostgreSQL with the given settings to verify if they are correct.

For more details (see Connections and Authentication).

JMS Settings

Warning: This part should not be modified ,unless you use a own HornetQ messaging server!



● Use embedded JMS Server ○ Use Specific JMS Server

JMS Server

Host: localhost

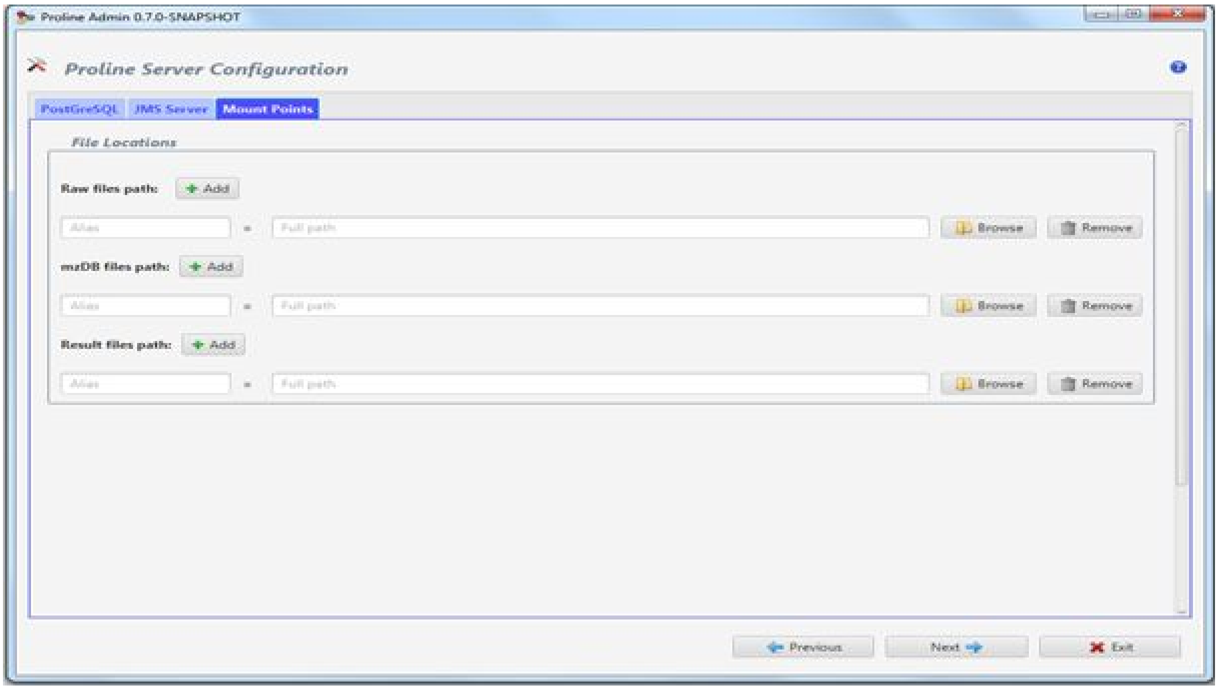
Port: 5445

Proline Queue Name: ProlineServiceRequestQueue

The communication between your local machine and the JMS Server is configured here. It is advised to use the embedded JMS server with its default values but if a JMS server has already been set up on another machine you can specify its address here.

- “Host”: the host name, its default value is “localhost”.
- “Port”: the port number, its default value is 5445.
- “Proline Queue Name”: the name of the queue that receives messages to a consumer, its default value is “ProlineServiceRequestQueue”.

Mount Points



File Locations

Raw files path: Alias = Full path

mzDB files path: Alias = Full path

Result files path: Alias = Full path

Proline Studio and Proline Web will be able to access the files from the locations provided here. Note that the section named *"Mount points"* is only enabled if the Proline server configuration file has been provided.
You can add as many mount points as you want. To do so, just select a path and a unique name for the mount point. The name will be visible by the Proline user so it should be explicit.
Three types of entries are available: Raw files, mzDB files and result files. The identification result files from Mascot, OMSSA, X!tandem and anothers must be defined in the *Result file path. **Proline users will only be able to import data from the "Result files" mount point.***
Note: When providing mount points paths, you can use slash ("/") or antislash ("\") separator. Both formats are valid in Proline-Admin GUI install.

PostgreSQL Server Configuration

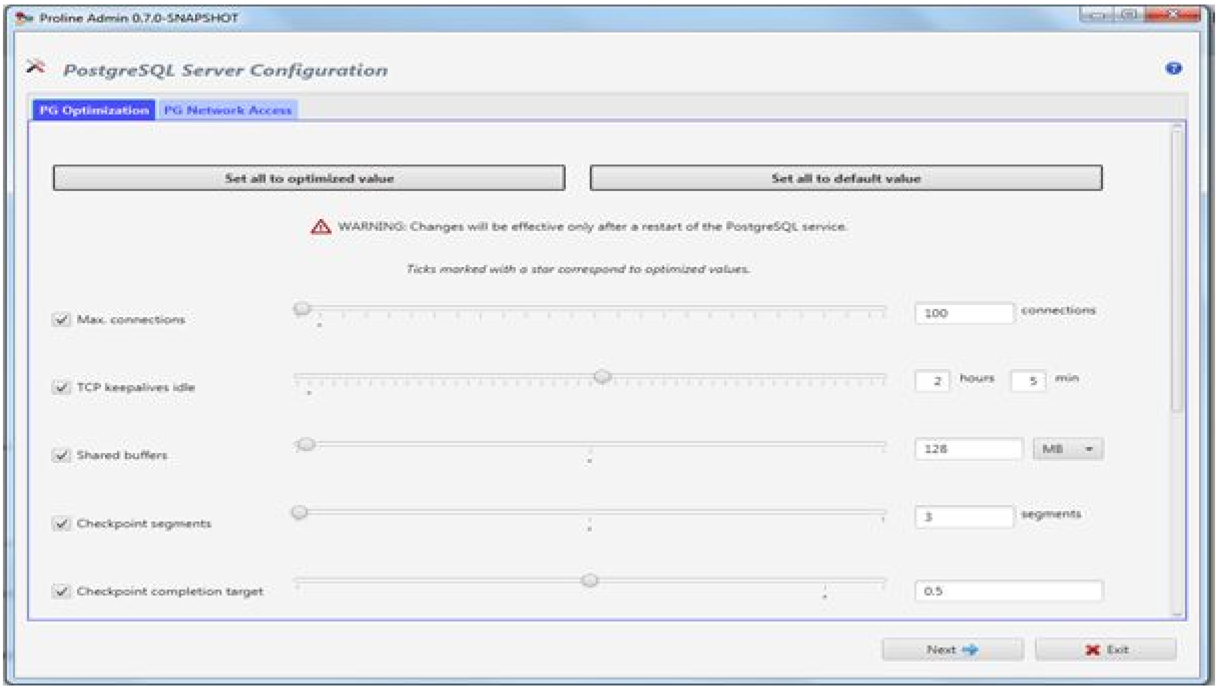
Warning: Configuring PostgreSQL requires administration rights or being super-user when running Proline-Admin GUI install. Also keep in mind that PostgreSQL service requires to be restarted to handle the changes made here.
This page will be displayed if the data directory has been defined in the first step.

PG Optimization

The PostgreSQL Optimization tab lets you modify the most important PostgreSQL settings. However, it is always possible for an advanced user to access all the parameters by editing the `postgresql.conf` file.
Two actions are available at the top of the window:

- "Set all to optimized value": compute for each parameter the most appropriate value given your machine specifications and Proline needs.
- "Set all to default values": reset each parameter to its default value. Note that these default values may reduce the PostgreSQL efficiency on a powerful server.

Each parameter can be unchecked, which means that it will be commented in the configuration file.
A tooltip is displayed when you hover a parameter with the mouse cursor.


Proline Admin 0.7.0-SNAPSHOT
PostgreSQL Server Configuration
PG Optimization
PG Network Access
Set all to optimized value
Set all to default value
Next
Exit

WARNING: Changes will be effective only after a restart of the PostgreSQL service.

Ticks marked with a star correspond to optimized values.

☑ Max. connections: 100 connections

☑ TCP keepalives idle: 2 hours 5 min

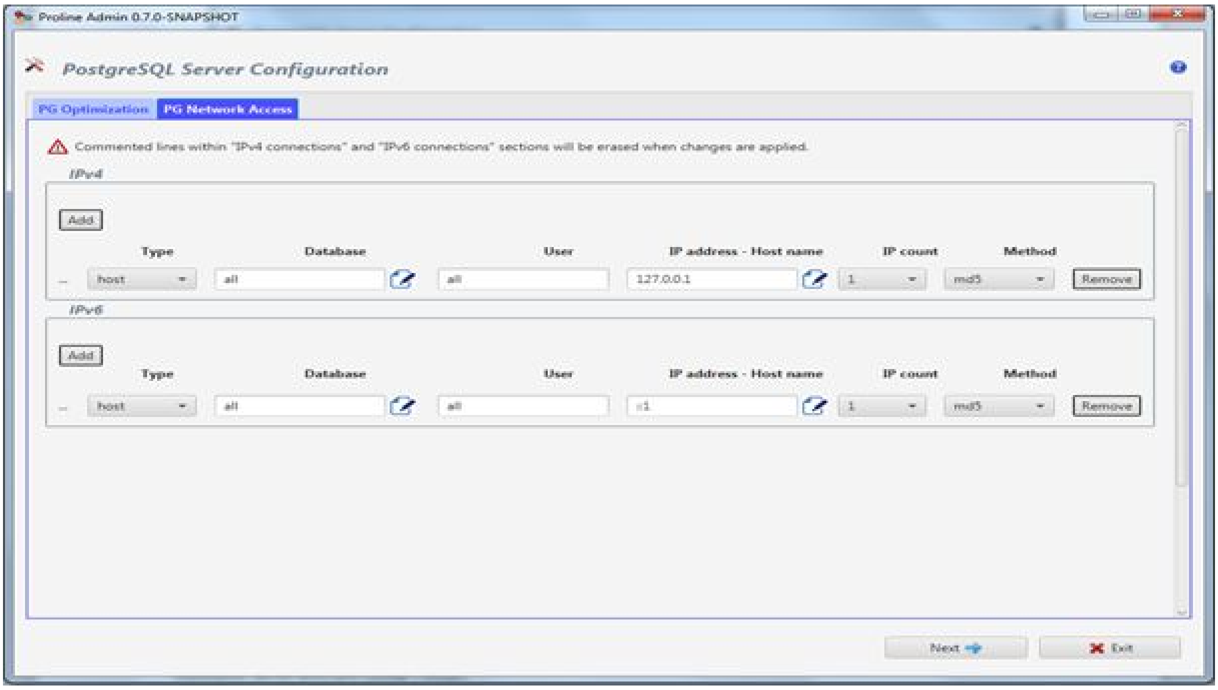
☑ Shared buffers: 128 MB

☑ Checkpoint segments: 3 segments

☑ Checkpoint completion target: 0.5

PG Access Right

The PG Access Rights tab lets you modify the PostgreSQL client authentication file `pg_hba.conf` easily. However it is always possible for an advanced user to access all the parameters by editing the `pg_hba.conf` file.
The default configuration is to allow all clients in your network, but you can restrain the access depending on your local network policy. Note that all Proline Studio clients need to access the PostgreSQL server.
You can give an access to a small selection of IP addresses by using the IP count. For instance, to accept connection for computers with IP addresses from 192.168.0.0 to 192.168.0.254 you can specify 192.168.0.0 as IP address and 255 as IP Count.



Proline Admin 0.7.0-SNAPSHOT

PostgreSQL Server Configuration

PG Optimization | PG Network Access

Commented lines within "IPv4 connections" and "IPv6 connections" sections will be erased when changes are applied.

IPv4

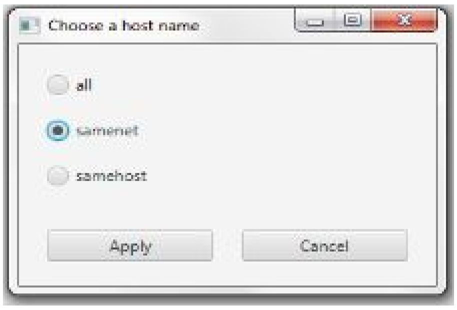
Type	Database	User	IP address - Host name	IP count	Method
host	all	all	127.0.0.1	1	md5

IPv6

Type	Database	User	IP address - Host name	IP count	Method
host	all	all	::1	1	md5

You can use the edit icon after the field “IP address/Host name” to open a window to help you choose a host name.

- “all”: to match any address.
- “samenet”: to match any address in any subnet that the server is directly connected to.
- “samehost”: to match only the clients on the current server.



Choose a host name

○ all
● samenet
○ samehost

The edit icon after the field “Database” opens a window to help you design the database access. See PostgreSQL Documentation for more help.

Proline Module Configuration

PostgreSQL

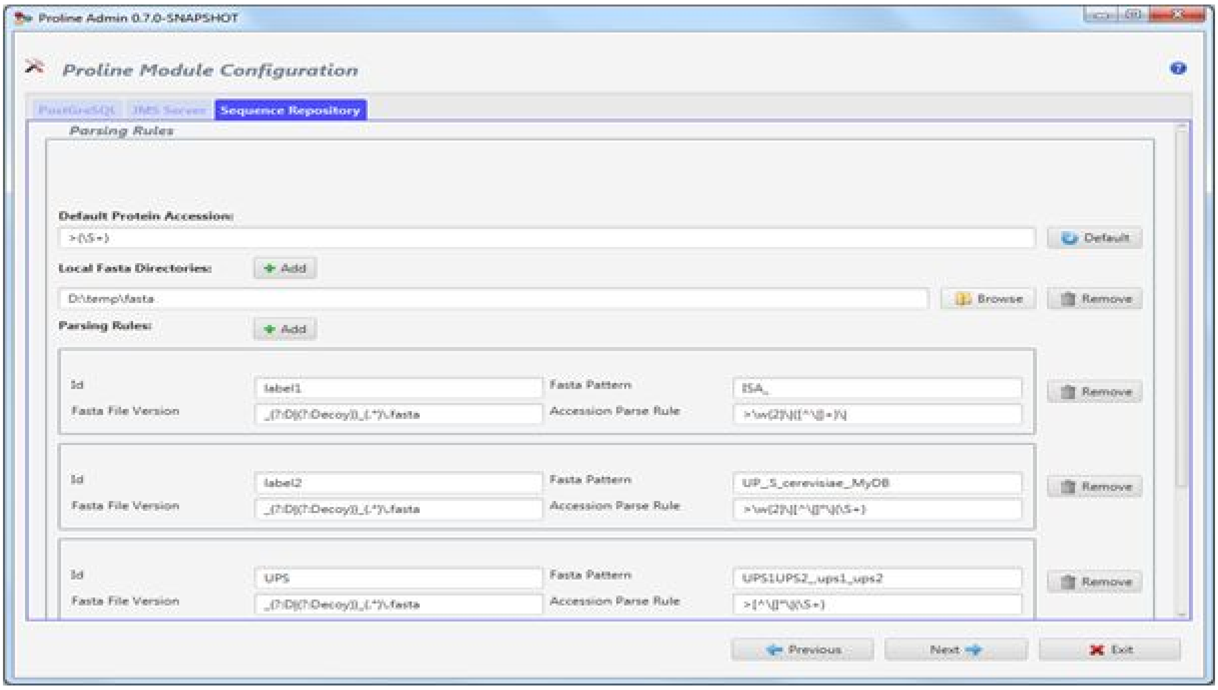
The PostgreSQL tab lets you edit the parameters to connect to the Proline databases. See chapter PostgreSQL for more details.

JMS Settings

The communication between your local machine and the JMS Server is configured here. See chapter JMS Settings for more details.

Sequence Repository

The Sequence Repository tab lets you edit the parsing rules used to capture the protein accession numbers in your Fasta files. These rules should match the rules that are used in search engines such as Mascot: the search engine will extract the accession number from a Fasta file to write it in its result file, and Proline will search for this accession number in the same Fasta file.



Default Protein Accession: >(\S+)

Local Fasta Directories: D:\temp\fasta

Parsing Rules:

Id: label1 — Fasta Pattern: ISA_
Fasta File Version: _(?:D|(?:Decoy))_(.*)\.fasta — Accession Parse Rule: >\w{2}\|([^\|]+)\|

Id: label2 — Fasta Pattern: UP_S_cerevisiae_MyDB
Fasta File Version: _(?:D|(?:Decoy))_(.*)\.fasta — Accession Parse Rule: >\w{2}\|[^\|]*\|(\S+)

Id: UPS — Fasta Pattern: UPS1UPS2_ups1_ups2
Fasta File Version: _(?:D|(?:Decoy))_(.*)\.fasta — Accession Parse Rule: >[^\|]*\|(\S+)

- “Default Protein Accession”: The default parsing rule to capture the protein accession number from a Fasta file. This parsing rule will be used on all Fasta files unless a specific parsing rule is defined below.

- The default value is « >(\\S+) »: It means that the text between the “greater than” sign and the first space character will be considered as the accession number. For instance, “sp|P10276|RARA_HUMAN” will be the accession number in the following Fasta line:
`>sp|P10276|RARA_HUMAN Retinoic acid receptor alpha OS=Homo sapiens OX=9606 GN=RARA PE=1 SV=2`

- The “Default” button will reset the Default Protein Accession to its default value.

- “Local Fasta Directories”: add one or more directories in which Fasta files are contained. The Sequence Repository will search for Fasta files in sub directories as well.

- "Parsing Rules": Parsing rules can be added for a specific Fasta files on which the default parsing rule would not work. It can be necessary if you have for instance different naming conventions for Uniprot and NCBI Fasta files, or for users who have different way to generate their files. A parsing rule requires the following information:
    - "Id ": a unique identifier for the parsing rule.
    - "Fasta pattern": this rule will be used on all the Fasta files matching this pattern. You can use regular expression but the pattern is case insensitive, so make sure that the pattern will not match unwanted Fasta files.
    - "Fasta File Version": a regular expression with capturing group to extract a release version from the Fasta file name (case insensitive).
    - "Accession number parsing rule": a regular expression with capturing group for the protein accession number.

For more details (see Installing and configuring the Sequence Repository)

Proline Web Configuration

Proline Web

Note: it is advised to use the default values of JMS server properties. If you wish to install Proline Web on a distinct server, you have to specify the Proline server host name.

Summary

This section provides a summary of everything you have configured so far. It lets you to check the new settings one last time before updating the configurations:

- Proline Server Configurations

"PostgreSQL: Ok" if the connection to the database successfully established otherwise "PostgreSQL: NOk"

"Mount Points": the number of mount points has been entered.

"JMS Server": The type of JMS server.

"Set up or Update Proline databases": the user should wait until Proline is setup or Proline Databases are updated.

Warning: this process can take a while.

- Proline Sequence repository

"PostgreSQL: Ok" if the connection to the database successfully established otherwise "PostgreSQL: NOk"

"Parsing rules": the number of parsing rules has been entered.

"JMS Server": The type of JMS server.

- Proline Web

"Mount Points": the number of entered mount points.

- PostgreSQL Server Configuration

Access Right and Optimization to PostgreSQL server.

Warning: Click on button "validate" will save all changes that have been entered in Proline components.

Installing Proline Server from archive files

First check that all requirements are first installed on the computer, then download the zip archive containing Proline components from the website (http://proline.profiproteomics.fr/download/).

The Proline Server archive file contains three others archives corresponding to the different components (Proline Server/ Sequence Repository/ Proline Admin). Unzip those components on the appropriated computer (Proline Server and Proline Admin are recommended to be on the same computer. Sequence Repository is recommended to be installed on the computer where the Fasta files are located).

When components are installed, you should setup the server and its component (see Setting up in next chapter).
Warning: Before running setup you should **create a Postgresql user** for Proline with "create database" rights. You can create a user using pgAdmin 3 (<pg_install_path>\PostgreSQL\9.4\bin\pgAdmin3.exe) or using createuser.exe in a command window (<pg_install_path>\PostgreSQL\9.4\bin\createuser.exe)

Setting up the Datastore and Proline Server

Once Proline configuration files have been edited. you must initialize Proline datastore and setup components using the Proline-Admin software provided with the Proline Suite. It is available as a command-line tool or as Graphical Interface called Proline Admin GUI install. You can either use the graphical interface Proline-Admin GUI install or use the command-line tool Proline-Admin to set up datastore.
Warning: Configuration files have to set up before using command line scripts.

From the command line

Set up the datastore by executing the following command:

on Windows systems

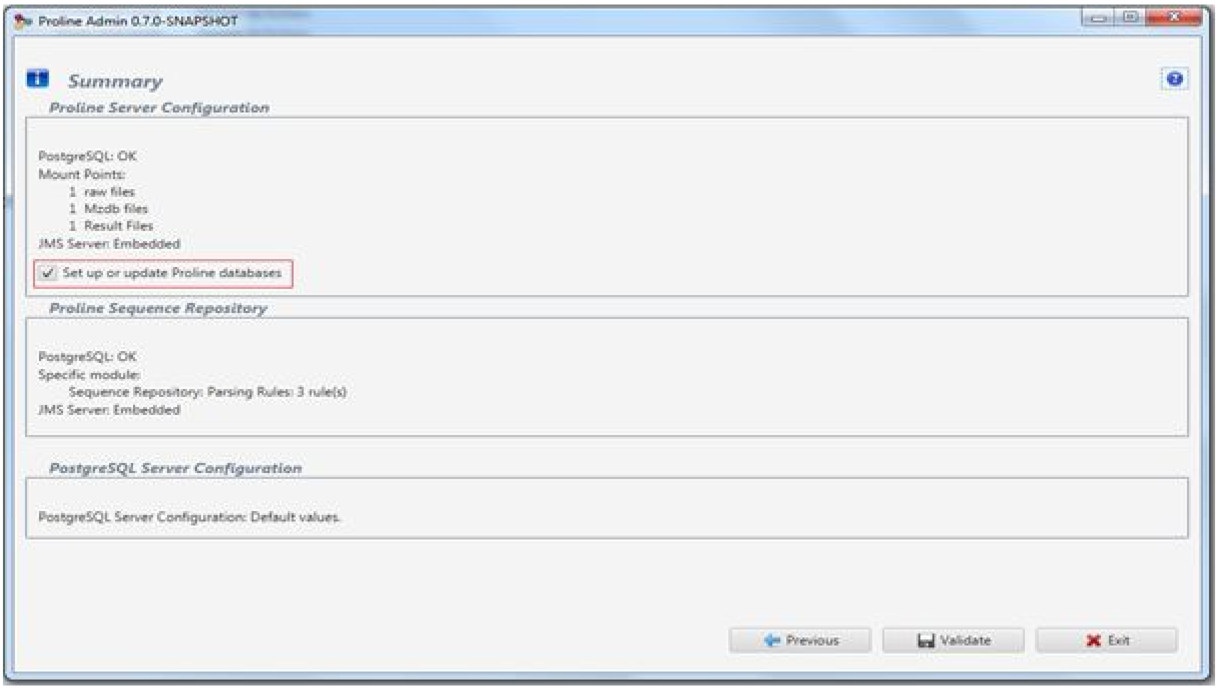
```
> run_cmd.bat setup
```

on Linux systems

```
> chmod +x run_cmd.sh
> run_cmd.sh setup
```

From the graphical Interface

Note : if you have not checked the button "Setup or update Proline databases" in first install of Proline. You can always use the graphical tool Proline-Admin GUI install to setup the datastore. You should just check the button "set up or update Proline databases" and validate.



Proline Server Configuration

PostgreSQL: OK
Mount Points:
    1 raw files
    1 Mzdb files
    1 Result Files
JMS Server: Embedded

☑ Set up or update Proline databases

Proline Sequence Repository

PostgreSQL: OK
Specific module:
    Sequence Repository: Parsing Rules: 3 rule(s)
JMS Server: Embedded

PostgreSQL Server Configuration

PostgreSQL Server Configuration: Default values.

Upgrading from a previous version

Upgrading Proline Server

If using the installer, there is no “upgrader”, so you must first uninstall the previous version of the server. The uninstallation will not remove previous data stored in Proline Datastore nor will it remove logs files.

Warning : Before doing so, to keep previous server configuration in order to copy it in new version, copy files <Proline_install>/server/config/*,<Proline_install>/admin/config/* and <Proline_install>/seqrepo/config/* in a safe folder. You can also keep the previous scripts in order to use the same settings in new ones.

Then you have to do a fresh install, using the same folder as previous installation. (See Installing Proline Server using Windows installer or Installing Proline server using archive files) . Once installation is done you can skip setup step and copy the saved configuration files in the fresh installation folders.

Updating the Datastore

To update the different Proline Databases, you need to get the appropriate version of Proline-Admin. You can use the command line or the provided graphical tool (Proline-Admin Gui monitor) see upgrade all databases.

From the command line

Update the datastore by executing the following command:

on Windows systems

```
> run_cmd.bat upgrade_dbs
```

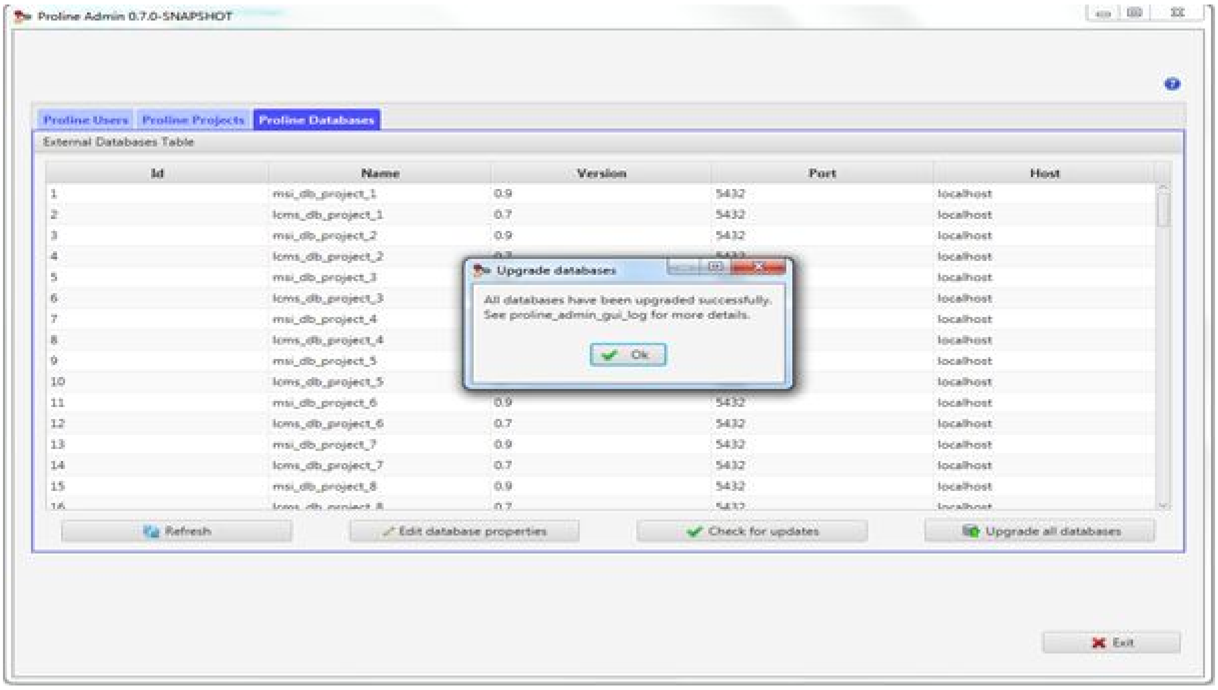
on Linux systems

```
> chmod +x run_cmd.sh
> run_cmd.sh upgrade_dbs
```

Verbose mode is used in order to display as much information as needed. You can view all displayed information in proline_admin_log.txt, and especially search for any “ERROR” message.

From the graphical interface

Open the graphical interface by running `monitor.bat` (on Windows) or `monitor.sh` (on linux system).
Note: If you don't have a valid configuration files, you can always change or edit them by using Proline-Admin GUI install (See Launch Proline-Admin GUI install).
Click on the button named Upgrade Proline databases.



Id	Name	Version	Port	Host
1	msi_db_project_1	0.9	5432	localhost
2	lcms_db_project_1	0.7	5432	localhost
3	msi_db_project_2	0.9	5432	localhost
4	lcms_db_project_2	0.7	5432	localhost
5	msi_db_project_3			localhost
6	lcms_db_project_3			localhost
7	msi_db_project_4			localhost
8	lcms_db_project_4			localhost
9	msi_db_project_5			localhost
10	lcms_db_project_5			localhost
11	msi_db_project_6	0.9	5432	localhost
12	lcms_db_project_6	0.7	5432	localhost
13	msi_db_project_7	0.9	5432	localhost
14	lcms_db_project_7	0.7	5432	localhost
15	msi_db_project_8	0.9	5432	localhost
16	lcms_db_project_8	0.7	5432	localhost

All databases have been upgraded successfully.
See proline_admin_gui_log for more details.

Note:

- This functionality is available only when you are connected to Proline databases (see how to configure and set up Proline, or using Proline Admin GUI).
- this action may take a while. Wait until this operation is finished.

Upgrading Sequence Repository

There is no upgrade procedure for this module. You should do a fresh install (see installing Sequence Repository) but you **can copy the previous configuration files in the fresh installation** !
You can always edit the configuration file via the graphical tool Proline-Admin GUI install then Proline Module Configuration finally the tab Sequence Repository; for more details see (Sequence Repository).

Upgrading Proline Studio

Proline Studio application distribution is a zip file that must be unzipped on each client PC. You should unzip it under a different folder than previous version. The executable file is in the “bin” directory.

Upgrading Proline Web Desktop

TODO

Setting up Proline with Proline-Admin GUI

Starting Proline-Admin GUI

If you used the Windows installer, a shortcut “Proline Admin” is available in the Windows Start Menu, under the Proline folder.

Otherwise, execute the `install.sh` (on Windows) or `install.bat` (on Linux) script located in the <Proline_installation_path>/Proline-Admin GUI folder.

Configure Proline Suite

If you have installed Proline with the installer, the configuration should have been done through the wizard .

Configure Proline parameters

See Launch Proline-Admin GUI install

Setting up Proline Datastore

Once the configuration files have been edited and validated via Proline-Admin GUI install, in the summary panel you should check the button *Set up or upgrade databases*. Click it to finish the Datastore setup (see Graphical Interface).

Setting up Proline manually

In this section the manual configuration is described for PostgreSQL and then for Proline-Admin as it must first be configured since this component is used to execute Proline commands (set up the datastore, creates projects and users).

PostgreSQL authorizations and optimization

The PostgreSQL configuration files are located in the “data” folder. By default, this folder is <PostgreSQL_install>/9.4/data.
First you must ensure that postgreSQL accept remote connections. The `postgresql.conf` file must be modified to allow remote computer connections. Edit the `listen_addresses` line to allow IP of remote computers. See PostgreSQL Documentation.
The `pg_hba.conf` file lists the IP addresses that are allowed to connect to your databases. In order to allow connections to the PostgreSQL server, make sure to add the following line (or more specific

mask if needed): `host all all 0.0.0.0/0 md5`. The configuration of `pg_hba.conf` is documented in PostgreSQL documentation.

Create a Postgresql user for Proline with “create database” rights.

PostgreSQL can affect the Proline Server performances; it is generally a good idea to optimize PostgreSQL parameters.

Most useful tuneable parameters in `postgresql.conf`

Following recommended memory sizes are given for a server with **16 GiB** of physical memory and about **8 GiB** dedicated for the PostgreSQL instance.

`max_connections`	Number of concurrent SQL sessions (each Proline Server task can use 1 to 5 SQL sessions, each Proline-Studio instance can use some SQL sessions). **Default value 100**
`tcp_keepalives_idle`	Number of seconds before sending a keepalive packet on an otherwise idle TCP connection. Help with broken router / firewall and checking for dead peers. **Default value 0 (2 hours) → 300 (5 minutes)**
`shared_buffers`	Use about 1/4 of physical memory dedicated to the PostgreSQL instance. **Default value 32 MB → 2048 MB**
`checkpoint_segments`	Use (`shared_buffers` / 16) ; max. 64 or 256 for write-heavy bulk loading. **Default value 3 → 128**
`checkpoint_completion_target`	0.9 for high value of `checkpoint_segments` **Default value 0.5 → 0.9**
`temp_buffers`	**Per session** Used for temporary tables creation. **Default value 8 MB → 512 MB**
`work_mem`	**Several per session** Used for hashing, sorting and `IN` operator when processing queries. **Default value 1 MB → 4 MB to 64 MB**
`maintenance_work_mem`	Used for initial index creation and `VACUUM` operations. **Default value 16 MB → 1024 MB**
`effective_cache_size`	Assumption about the effective size of the disk cache to optimize index use (Monitor physical memory allocated by system to disk cache operations). **Default value 128 MB → 4096 MB**

Setting up Proline Datastore

Edit the configuration file `config/application.conf` located in the Proline Admin folder. Then perform the datastore setup by running the dedicated script.

The Proline Admin program files are located in the “.\Admin-GUI” sub-folder of the Proline installation directory if you used the Windows installer. Otherwise they should be in the folder obtained after Proline Admin GUI archive file extraction.

Configuration files

To allow Proline Admin to access to Proline Datastore, you should modify the `application.conf` file located under the config folder of your Proline Admin installation directory.

```
server-config-file = ""
pwx-config-file = ""
postgresql-data-dir = ""

proline-config {
 driver-type = "postgresql" // valid values are: h2, postgresql or sqlite
 data-directory = "/Path/to/Proline/Data" //Not used actually...
}

auth-config {
 user="<user-proline>" //!! SET TO Database Proline user login
 password="<proline_user_password>" //!! SET TO Database Proline user password
}

host-config {
 host="your_postgresql_server_host" //!! Do NOT put "localhost", but the real IP
address or fully qualified name
 port="5432" //or other port used to access your DBMS
}
```

Note:

- `auth-config`: specify the PostgreSQL Proline user (and its password). This user should have been created when installing/setting up PostgreSQL.
- `host-config`: IP adress or name of the server hosting PostgreSQL. Don’t use localhost!
- default naming scheme of databases created by Proline can be modified by editing `config/application-<dbtype>.conf` file. But it should not be modified if not necessary.
- Other parameters should not be modified unless specific usage.
- `postgresql-data-dir`, `server-config-file` and `pwx-config-file` may only be used when using Admin GUI configuration editor

Do not confuse:

- 'data-directory' under 'proline-config': indicates where to store Proline data (e.g. exports).
- 'postgresql-data-dir': references the PostgreSQL data directory (for Proline-Admin purposes).

Setting up Proline Datastore

Once all configuration has been done, run `setup_proline.bat` (on Windows) or `setup_proline.sh` (on Linux) from the Proline-Admin installation directory. This command will create the required databases in the Datastore.

Proline Server

When using command line configuration, you should manually edit and configure Proline Server file `config/application.conf`. The server folder is located in the “.\Server_Cortex” subfolder if the Windows installer has been used or in the folder in which the “Proline Server” archive has been unzipped.

Configure the Datastore

Edit the `application.conf` file in the same way you did it for Proline Admin (see Admin Configuration files). If your configuration is valid, the Proline Server will be able to use the datastore you've created using Proline Admin.

Configure the mount-points

Result identification files (Mascot, OMSSA or X!Tandem) as well as mzDB files (for the XIC Quantitation process) are only browsed from Proline Server side.

Warning: on a Windows server installation, please use UNC style to access remote disks instead of mounted disk on letters. Otherwise Proline won't be fully functional. ex: use \\\\my_server\\disk\\folder instead of a mounted disk on Z: (Z: mounted as \\my_server\\disk\\folder)

Mascot or OMSSA path should be configured in `result_files` sub-entry, administrator can add one or more mappings as `label = “<absolute/directory/path>”`. mzDB files path should be set under `mzdb_files` sub-entry.

Label can be any valid string chosen by Administrator to help user identify mount_point. If multiple repositories are defined, labels must be different.

Configuration examples:

```
mount_points {
 result_files {
        mascot_data = "Z:/" //under window environment
        omssa_data = "/local/omssa/data" //under linux environment
        xtandem_data = "U:/"
        ...
 }
 ...
 mzdb_files {
 }
}
```

Configure the Messaging connection

Unless you use a own HornetQ messaging server, this file should not be modified !
If needed, edit the `jms-node.conf` file to specify : the server hosting the messaging server (`jms_server_host`), the name of the Proline Queue (`proline_service_request_queue_name;` **warning**: should also be modified on client side), the number of thread Proline server could use (`service_thread_pool_size;` *-1* let the server define the appropriated value)

```
node_config {
  jms_server_host = "localhost"
  jms_server_port = 5445
  proline_service_request_queue_name = "ProlineServiceRequestQueue"
  service_thread_pool_size = -1
  xic_files_pool_size = 2
  enable_imports = true
}
```

Running Proline Server

Running the Server

Server Memory Setup

The default amount of memory used by the server can be changed in the `start_cortex.bat` (`start_cortex.sh` on Linux) file. If the server is configured with large amount of memory, it is recommended to increase this value. Change the value of the `–Xmx` java option to `–Xmx9g` to increase the available memory to 9Gb.

Running on Windows

On Windows, the server is started from the command : `start_server.bat`.
(This single command will start HornetQ then Cortex by calling successively `start_HornetQ.bat` and `start_cortex.bat`).

Running on Linux

On Linux, the server is started by running `start_HornetQ.sh` first then `start_cortex.sh`.

Note: All these scripts are in the directory <Server_Installation_Path>/server/

Proline-Admin GUI monitor

Proline administrator can manage Proline users, projects and external databases via a graphical interface tool called Proline-Admin GUI monitor.
This graphical tool helps the administrator to realize several operations for example:
Add a Proline user, disable a Proline user, activate a Proline user, create a new project, activate a project, disable a project, archive an existing project and restore an archived project …

Launch Proline-Admin GUI monitor

The graphical interface can be started by double clicking on the starting scripts:

- `monitor.bat` (on Windows).
- `monitor.sh` (on Linux).

Note:
All these scripts are in the directory <Server_Installation_Path>/Proline-Admin-GUI/
The scripts will launch the application and you will find your logs under the folder logs.

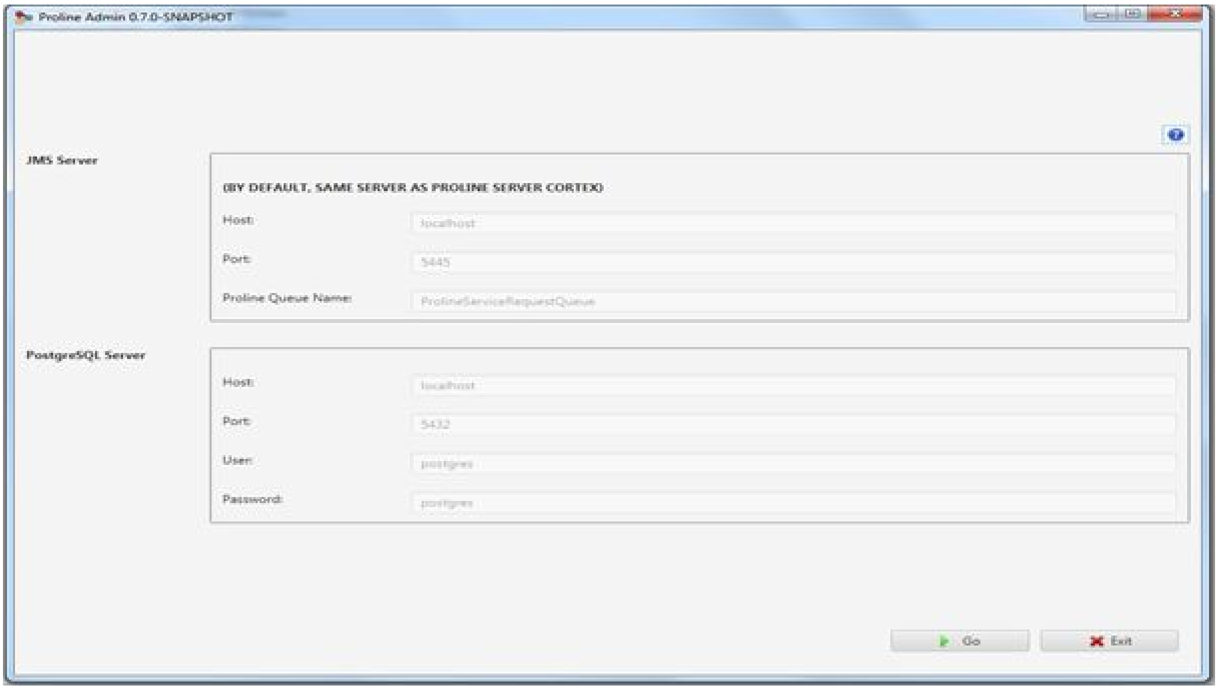
The graphical interface in details

Warning: Proline-Admin GUI monitor will not work if you are not connected to the database server. Before using the application make sure that you have already setup Proline configuration files and the datastore.

Proline properties summary

This view summarizes:

- JMS Server: the properties used in your JMS Server. It is advised to use the default values. If you have not setup Proline and the path of Proline server configuration file not found or empty, thus the default values will be used.
- PostgreSQL Server: the properties used in Proline-Admin to connect to the databases.



JMS Server

(BY DEFAULT, SAME SERVER AS PROLINE SERVER CORTEX)

Host: localhost

Port: 5445

Proline Queue Name: ProlineServiceRequestQueue

PostgreSQL Server

Host: localhost

Port: 5432

User: postgres

Password: postgres

If Proline has been installed properly and you are connected to the database server. Click on the go button to start using Proline-Admin GUI monitor. You will have a general view with Proline users, Proline projects and Proline external databases.

Proline user management

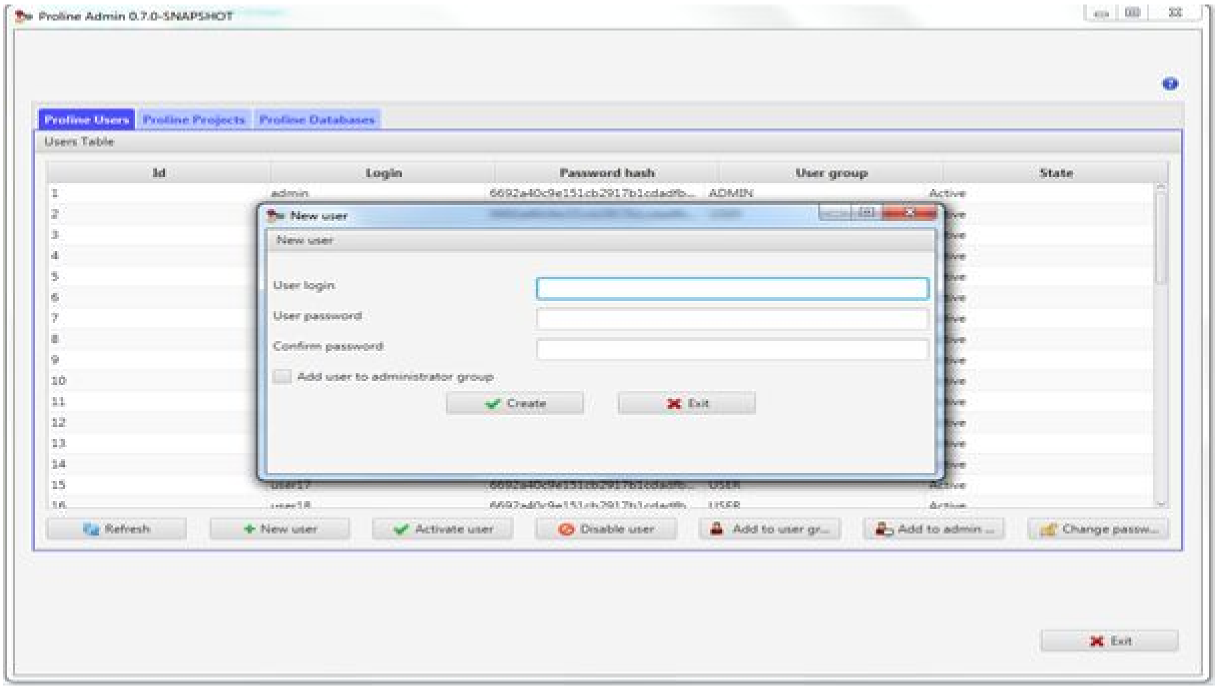
From the graphical tool Proline-Admin GUI monitor

Create a new Proline user

Click on the button New user, you must fill the required fields.

- Login: the identifier for the new account. You must choose a login that does not exist.
- Password: the user password.
- Confirm password: both passwords must be the same.
- Add user to the Proline administrator group: it adds this user to the Proline administrator group.

Note: standard users should not be in the administrator group.



Users Table

Id	Login	Password hash	User group	State
1	admin	6692a40c9e151cb2917b1cdadfb...	ADMIN	Active
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15	user17	6692a40c9e151cb2917b1cdadfb...	USER	Active
16	user18	6692a40c9e151cb2917b1cdadfb...	USER	Active

New user

User login

User password

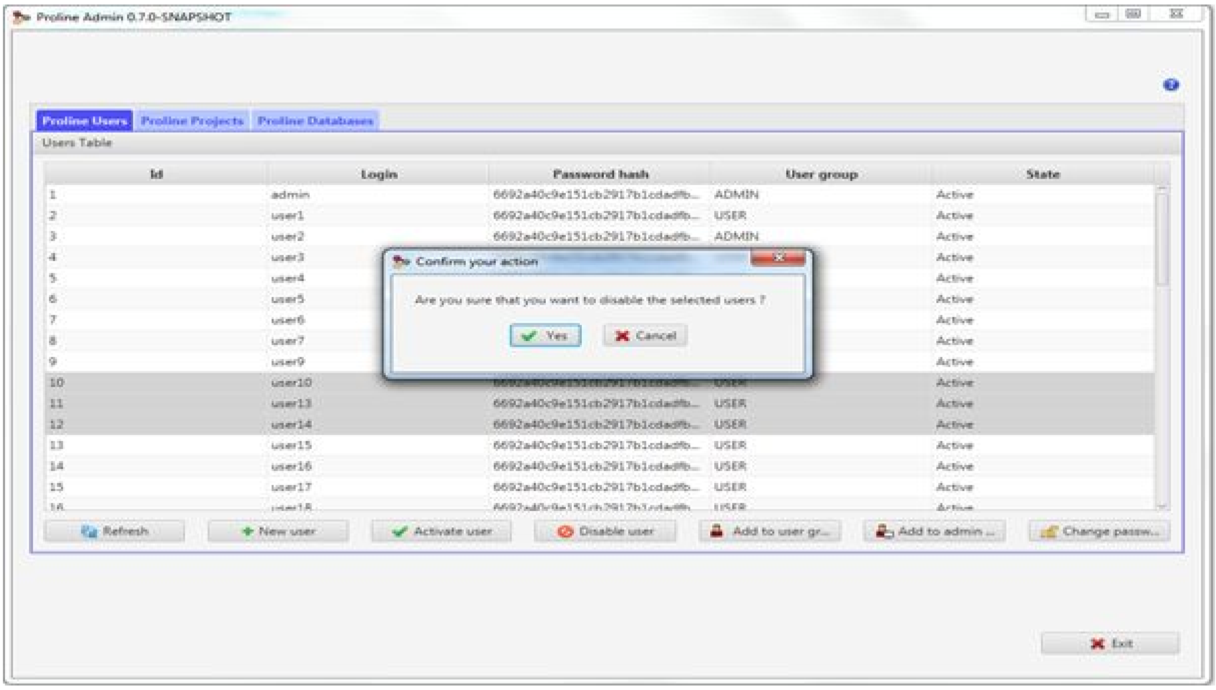
Confirm password

□ Add user to administrator group

Activate/Disable a Proline user

This action will activate/disable the selected Proline user(s). The user will not be deleted from the uds_db database, the account is just disabled.

Note: You can reactivate this user later.



Users Table

Id	Login	Password hash	User group	State
1	admin	6692a40c9e151cb2917b1cdadfb...	ADMIN	Active
2	user1	6692a40c9e151cb2917b1cdadfb...	USER	Active
3	user2	6692a40c9e151cb2917b1cdadfb...	ADMIN	Active
4	user3			Active
5	user4			Active
6	user5			Active
7	user6			Active
8	user7			Active
9	user9			Active
10	user10		USER	Active
11	user13	6692a40c9e151cb2917b1cdadfb...	USER	Active
12	user14	6692a40c9e151cb2917b1cdadfb...	USER	Active
13	user15	6692a40c9e151cb2917b1cdadfb...	USER	Active
14	user16	6692a40c9e151cb2917b1cdadfb...	USER	Active
15	user17	6692a40c9e151cb2917b1cdadfb...	USER	Active
16	user18	6692a40c9e151cb2917b1cdadfb...	USER	Active

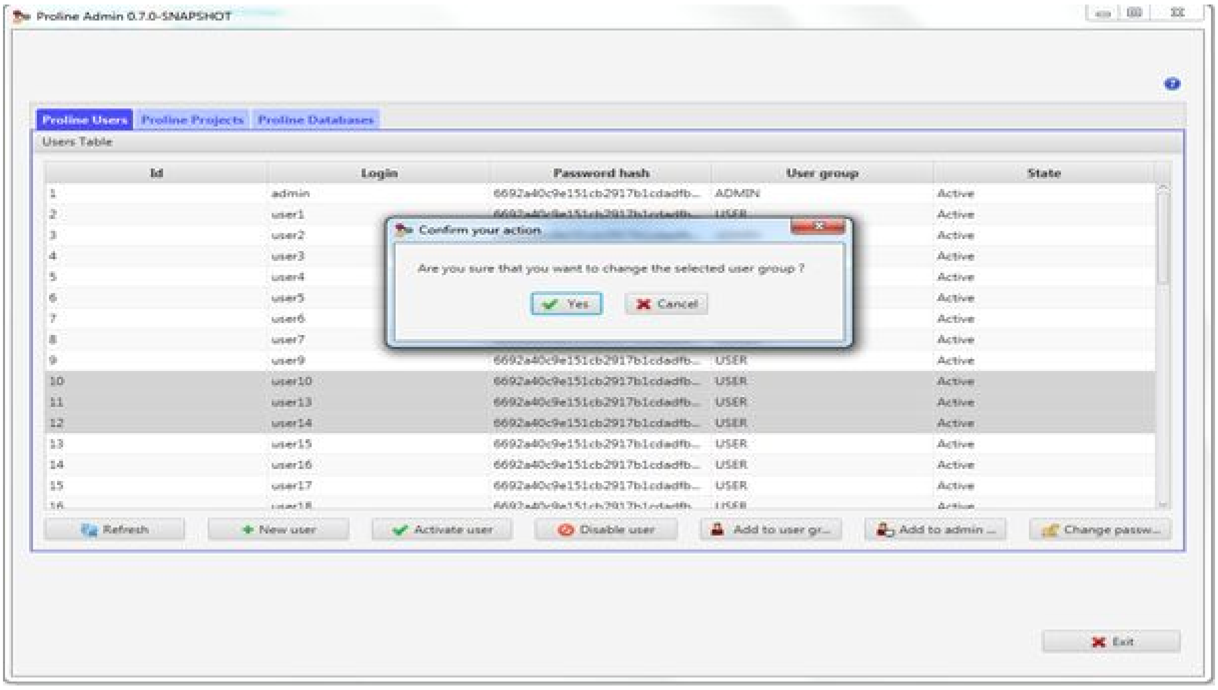
Confirm your action

Are you sure that you want to disable the selected users ?

Add to User group/Admin group

This action will add the selected user(s) to admin group or to user group.

Note: standard Proline users should not be in the Admin group.



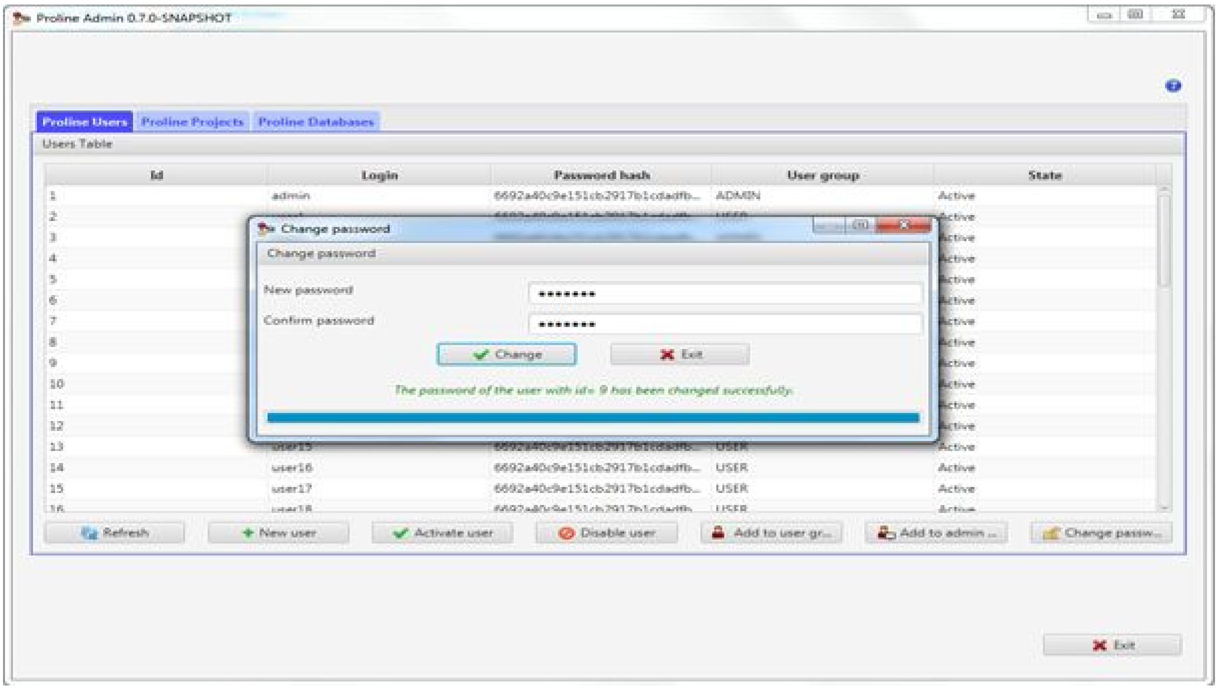
Id	Login	Password hash	User group	State
1	admin	6692a40c9e151cb2917b1cdadfb...	ADMIN	Active
2	user1			Active
3	user2			Active
4	user3			Active
5	user4			Active
6	user5			Active
7	user6			Active
8	user7			Active
9	user9	6692a40c9e151cb2917b1cdadfb...	USER	Active
10	user10	6692a40c9e151cb2917b1cdadfb...	USER	Active
11	user13	6692a40c9e151cb2917b1cdadfb...	USER	Active
12	user14	6692a40c9e151cb2917b1cdadfb...	USER	Active
13	user15	6692a40c9e151cb2917b1cdadfb...	USER	Active
14	user16	6692a40c9e151cb2917b1cdadfb...	USER	Active
15	user17	6692a40c9e151cb2917b1cdadfb...	USER	Active
16	user18	6692a40c9e151cb2917b1cdadfb...	USER	Active

Confirm your action

Are you sure that you want to change the selected user group ?

Change Password

This action will change the password for the selected Proline user. The entered passwords must match. If the password fields are empty, hence the default password will be used is “proline”.



Id	Login	Password hash	User group	State
1	admin	6692a40c9e151cb2917b1cdadfb...	ADMIN	Active
2				ctive
3				ctive
4				ctive
5				ctive
6				ctive
7				ctive
8				ctive
9				ctive
10				ctive
11				ctive
12				ctive
13	user15	6692a40c9e151cb2917b1cdadfb...	USER	Active
14	user16	6692a40c9e151cb2917b1cdadfb...	USER	Active
15	user17	6692a40c9e151cb2917b1cdadfb...	USER	Active
16	user18	6692a40c9e151cb2917b1cdadfb...	USER	Active

Change password

Change password

New password: ••••••••

Confirm password: ••••••••

The password of the user with id= 9 has been changed successfully.

Note:

- The grid can be sorted on one or many columns. Click on a column header to make it the sorting feature. Shift-click on another to activate multi-sorting.

From the Proline Admin command line

Create Proline user

You can create a Proline user with the Proline Admin "RunCommand" script. Open a command line window and type the following command:

- Windows: `run_cmd.bat create_user -l <user_login> -p <user_password> -a <administrator>`

- Linux: `sh run_cmd.sh create_user -l <user_login> -p <user_password> -a <administrator>`

Note: The password is optional. If not provided, the default password will be "proline".

If you specify -a the user will be in administrator group otherwise in user group.

Activate/disable Proline user

- Windows: `run_cmd.bat change_user_state -uid <user_id> -d <disable>`

- Linux: `sh run_cmd.sh change_user_state -uid <user_id> -d <disable>`

Note: If you specify -d (optional) the user will be disabled otherwise he will be activated.

change user group

- Windows: `run_cmd.bat change_user_group -uid <user_id> -a <administrator>`

- Linux: `sh run_cmd.sh change_user_group -uid <user_id> -a <administrator>`

Note: If you specify -a (optional) the user will be in administrator group otherwise in user group.

change user password

- Windows: `run_cmd.bat change_password -uid <user_id> -np <new_password>`

- Linux : `sh run_cmd.sh change_password -uid <user_id> -np <new_password>`

Note: The password is optional. If not provided, the default password will be "proline".

Proline project management

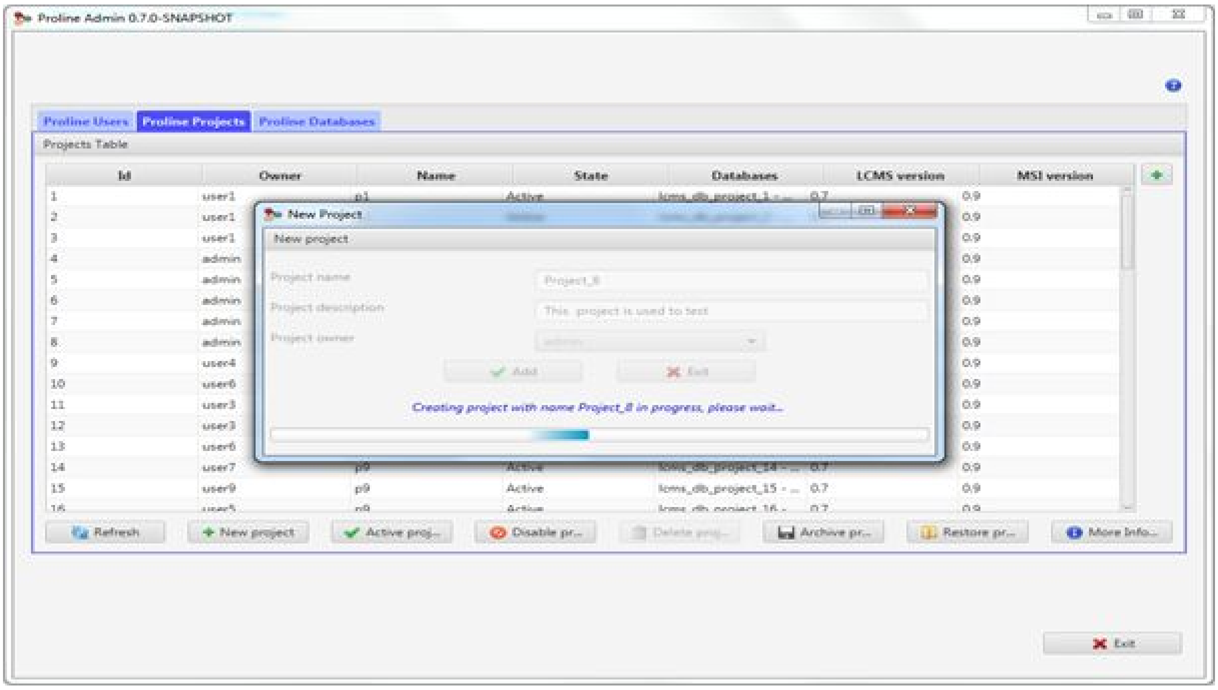
From the graphical tool Proline-Admin GUI monitor

New Project

This action will create a new Proline project. You must fill the required fields.

- Project name: the project name must not be empty.
- Project description: The project description is optional.
- Project owner: The project owner must not be empty. You can select the owner from the list of users.

Warning: a progress bar will appear until the action of creating a new project will finish.



Id	Owner	Name	State	Databases	LCMS version	MSI version
1	user1	p1	Active	lcms_db_project_1 - ...	0.7	0.9
2	user1					0.9
3	user1					0.9
4	admin					0.9
5	admin					0.9
6	admin					0.9
7	admin					0.9
8	admin					0.9
9	user4					0.9
10	user6					0.9
11	user3					0.9
12	user3					0.9
13	user6					0.9
14	user7	p9	Active	lcms_db_project_14 - ...	0.7	0.9
15	user9	p9	Active	lcms_db_project_15 - ...	0.7	0.9
16	user5	p9	Active	lcms_db_project_16 - ...	0.7	0.9

New project

Project name: Project_8

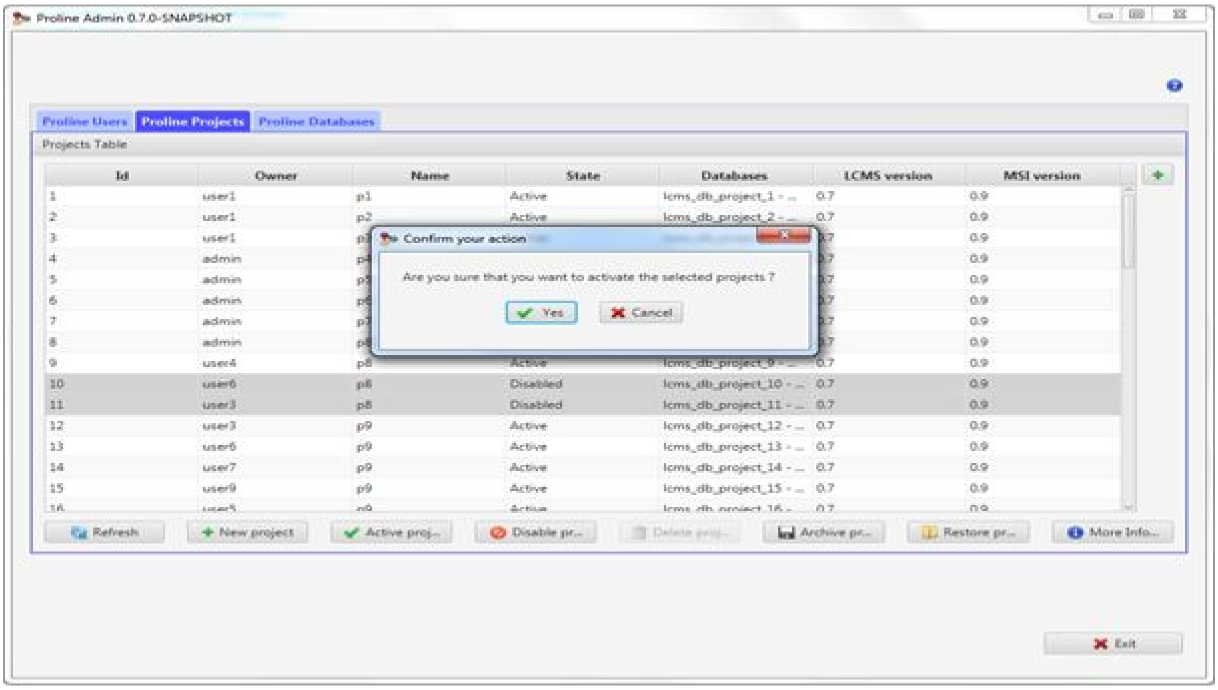
Project description: This project is used to test

Project owner: admin

Creating project with name Project_8 in progress, please wait...

Activate/Disable Project

This action will activate/disable the selected Proline project(s). The selected projects will not be deleted and can be re-activated later.



Id	Owner	Name	State	Databases	LCMS version	MSI version
1	user1	p1	Active	lcms_db_project_1 - ...	0.7	0.9
2	user1	p2	Active	lcms_db_project_2 - ...	0.7	0.9
3	user1	p3			0.7	0.9
4	admin	p4			0.7	0.9
5	admin	p5			0.7	0.9
6	admin	p6			0.7	0.9
7	admin	p7			0.7	0.9
8	admin	p8			0.7	0.9
9	user4	p8	Active	lcms_db_project_9 - ...	0.7	0.9
10	user6	p8	Disabled	lcms_db_project_10 - ...	0.7	0.9
11	user3	p8	Disabled	lcms_db_project_11 - ...	0.7	0.9
12	user3	p9	Active	lcms_db_project_12 - ...	0.7	0.9
13	user6	p9	Active	lcms_db_project_13 - ...	0.7	0.9
14	user7	p9	Active	lcms_db_project_14 - ...	0.7	0.9
15	user9	p9	Active	lcms_db_project_15 - ...	0.7	0.9
16	user5	p9	Active	lcms_db_project_16 - ...	0.7	0.9

Are you sure that you want to activate the selected projects ?

Note: you can select one or a list of projects to activate/disable them in batch.

Delete Project

This action will delete the selected Proline project.

Warning: This action will delete permanently the selected Proline project. It will delete physically the selected project and its external databases.

Archive project

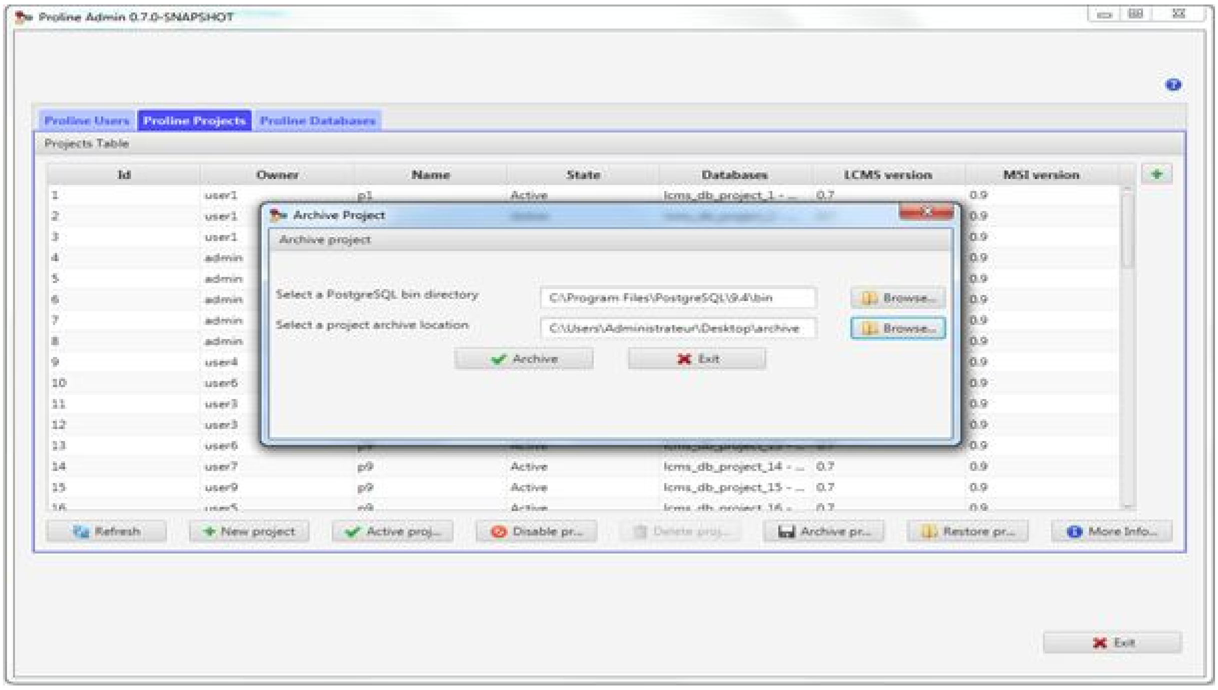
This action allows Proline administrator to archive the selected Proline project in a zip folder. The archived project is a copy of the project at the moment when the task of archiving project starts.

Note: The archived project in a zip folder can be shared with other sites, or attribute a Proline project to another Proline user on other database server.

The archived project folder contains 3 main files:

1. lcms_db_project_N.bak: a pg_dump of lcms_project_N database.
2. msi_db_project_N.bak : a pg_dump of msi_db_project_N database.
3. project_properties.json : a json file containing all metadata for project N: project user ,the version number, datasets, run identification, etc.

- Select PostgreSQL bin directory: the bin directory must contain the pg_dump executable that will be used to dump databases.
- Select a project archive location: the location where the project will be archived.
- Archive button: start archiving project task.



Id	Owner	Name	State	Databases	LCMS version	MSI version
1	user1	p1	Active	lcms_db_project_1 - ...	0.7	0.9
2	user1					0.9
3	user1					0.9
4	admin					0.9
5	admin					0.9
6	admin					0.9
7	admin					0.9
8	admin					0.9
9	user4					0.9
10	user6					0.9
11	user3					0.9
12	user3					0.9
13	user6	p9	Active			0.9
14	user7	p9	Active	lcms_db_project_14 - ...	0.7	0.9
15	user9	p9	Active	lcms_db_project_15 - ...	0.7	0.9
16	user5	p9	Active	lcms_db_project_16 - ...	0.7	0.9

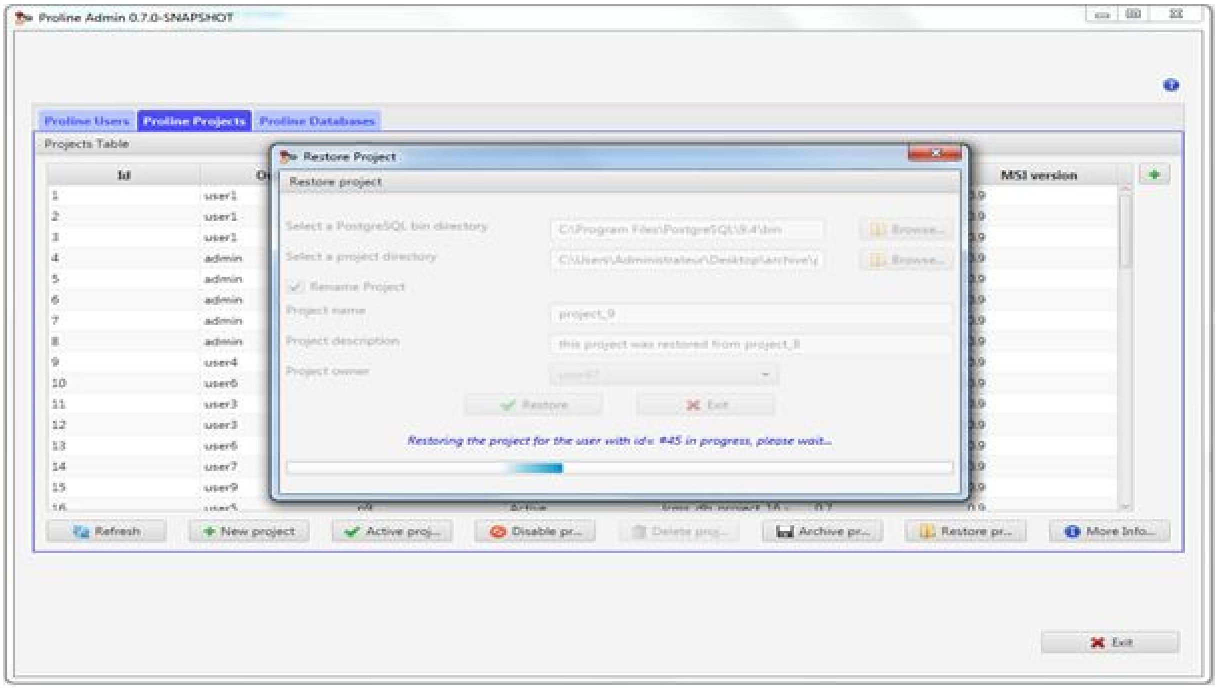
Select a PostgreSQL bin directory: C:\Program Files\PostgreSQL\9.4\bin

Select a project archive location: C:\Users\Administrateur\Desktop\archive

Restore project

This action allows Proline administrator to restore an archived project. This action will attribute the archived project to an existing user in Proline databases.

- Select a PostgreSQL bin directory: the bin directory must contain the pg_restore executable that will be used to restore Proline databases.
- Select a project directory: the project directory should contain the required files to restore a Proline project: lcms_db_project_N.bak, msi_db_project_N.bak and project_properties.json.
- Rename project: Import the archived project with another name. this can be necessary if the selected user already has a project with the same name (project names must be unique).
- Project description: optional project description.
- Project owner: the restored project will be attributed to the selected user from the list.



Select a PostgreSQL bin directory: C:\Program Files\PostgreSQL\9.4\bin

Select a project directory: C:\Users\Administrateur\Desktop\archive\

☑ Rename Project

Project name: project_9

Project description: this project was restored from project_8

Project owner: user67

Restoring the project for the user with id= #45 in progress, please wait...

Note:

- The archive and restore project actions can take hours depending on the project size.
- Be careful when restoring projects from another site that Proline server and PostgreSQL versions are the same.

More Info

This button is used to show a specific information about the selected project.

Note:

- The grid can be sorted on one or many columns. Click on a column header to make it the sorting feature. Shift-click on another to activate multi-sorting.
- This functionality is only available when Proline is set up (see how to setup Proline).
- The button ✚ will extend the projects table to show database's size (LCMS and MSI databases).

From the Command line

create a new Project

Run the following command line from the Proline-Admin directory. Open a command line window and run :

- Windows: `run_cmd.bat create_project -oid <owner_id> -n <project_name> -desc <project_description>`

- Linux: `sh run_cmd.sh create_project -oid <owner_id> -n <project_name> -desc <project_description>`

Note:

- The project description is optional.
- Project creation can also be done from Proline Studio.

Activate/disable a project

Run the following command line from the Proline-Admin directory. Open a command line window and run :

- Windows: `run_cmd.bat change_project_state -pid <project_id> -d <disable>`
- Linux: `sh run_cmd.sh change_project_state -pid <project_id> -d <disable>`

Note: Use -d to disable the project, do not use -d to activate it.

Archive project

Run the following command line from the Proline-Admin directory. Open a command line window and run:

- Windows: `run_cmd.bat archive_project -ad <archive_directory_path> -bd <pg_bin_directory_path> -pid <project_id>`
- Linux: `sh run_cmd.sh  archive_project -ad <archive_directory_path> -bd <pg_bin_directory_path> -pid <project_id>`

Example: to archive a Proline project with id=1 in the directory “C:\Users\Administrateur\archive” and PostgreSQL bin directory path is “C:\Program Files\PostgreSQL\9.4\bin” using Windows.

```
>run_cmd.bat archive_project -ad "C:\Users\Administrateur\archive" -bd
"C:\Program Files\PostgreSQL\9.4\bin” -pid 1
```

warning:

- Make sure that you are connected to the database server.
- Make sure that you are allowed to write in the archive directory.
- Make sure that the PostgreSQL bin directory contains pg_dump.exe.
- This action might take a while depending on the archived project size.

Restore project

Run the following command line from the Proline-Admin directory. Open a command line window and run :

- Windows : `run_cmd.bat restore_project -pd <project_directory_path> -bd <pg_bin_directory_path> -pid <project_id> -oid <project_owner> -n <project_name>`
- Linux : `sh run_cmd.sh  restore_project -pd <project_directory_path> -bd <pg_bin_directory_path> -pid <project_id> -oid <project_owner> -n <project_name>`

warning:

- Make sure that you are connected to the database server.

- Make sure that you are allowed to read from the archive directory.
- Make sure that the project name is unique for this user.
- Be careful when restoring projects from another site that Proline server and PostgreSQL versions are the same (tested with PostgreSQL version above 9.4)
- This action might take a while depending on the restored project size.

Proline databases management

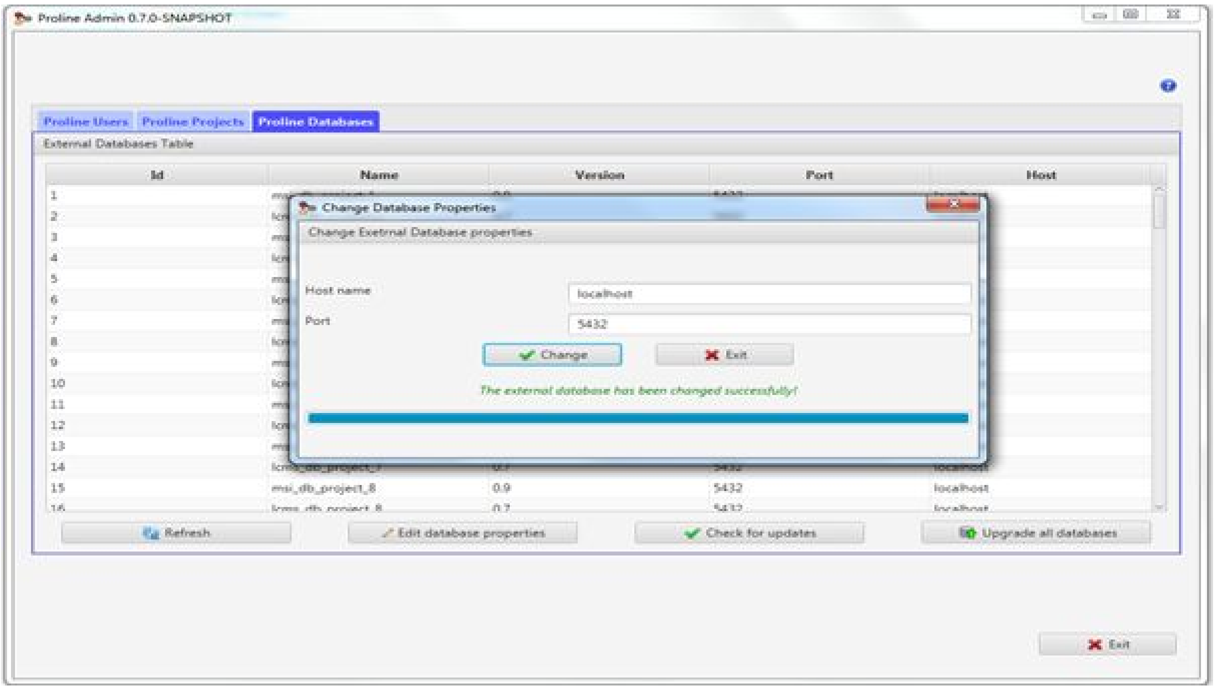
From the graphical tool Proline Admin GUI monitor

Refresh

This action will refresh and synchronize the table view with the existing data from the database.

Edit database properties

This action allows Proline administrator to change the connection properties of the selected database(s). Only host name and port number can be updated.
Warning: the database connection properties must not be empty.



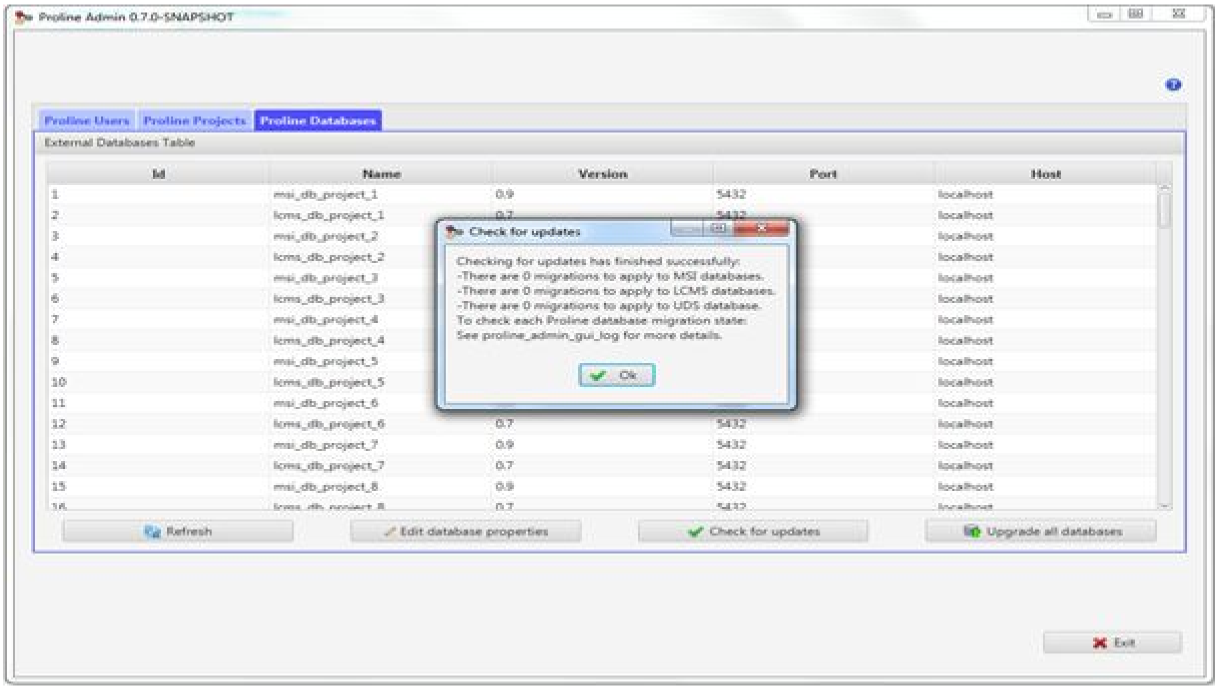
Id	Name	Version	Port	Host
1				
2	lc			
3	m			
4	lc			
5	m			
6	lc			
7	m			
8	lc			
9	m			
10	lc			
11	m			
12	lc			
13	m			
14	lcms_db_project_7	0.7	5432	localhost
15	msi_db_project_8	0.9	5432	localhost
16	lcms_db_project_8	0.7	5432	localhost

Host name: localhost
Port: 5432

The external database has been changed successfully!

Check for updates

This action will check for available updates. It will check the state of all projects. It summarize the number of migrations to apply to the databases.
Note: This action may take several minutes.



Id	Name	Version	Port	Host
1	msi_db_project_1	0.9	5432	localhost
2	lcms_db_project_1	0.7	5432	localhost
3	msi_db_project_2			localhost
4	lcms_db_project_2			localhost
5	msi_db_project_3			localhost
6	lcms_db_project_3			localhost
7	msi_db_project_4			localhost
8	lcms_db_project_4			localhost
9	msi_db_project_5			localhost
10	lcms_db_project_5			localhost
11	msi_db_project_6			localhost
12	lcms_db_project_6	0.7	5432	localhost
13	msi_db_project_7	0.9	5432	localhost
14	lcms_db_project_7	0.7	5432	localhost
15	msi_db_project_8	0.9	5432	localhost
16	lcms_db_project_8	0.7	5432	localhost

Checking for updates has finished successfully:
-There are 0 migrations to apply to MSI databases.
-There are 0 migrations to apply to LCMS databases.
-There are 0 migrations to apply to UDS database.
To check each Proline database migration state:
See proline_admin_gui_log for more details.

Upgrade all databases

This action will upgrade all databases to the last version.
Note: This action will take a while. You must wait until the migrations are applied to all databases.

From the Command line

check_for_updates

Run the following command line from the Proline-Admin directory. Open a command line window and run:

- Windows: `run_cmd.bat check_for_updates`
- Linux: `sh run_cmd.sh  check_for_updates`

Note :

- This action will check the state of all databases and check for available updates without installing any.
- This action can take several minutes.

upgrade_dbs

Run the following command line from the Proline Admin directory. Open a command line window and run:

- Windows: `run_cmd.bat upgrade_dbs`
- Linux: `sh run_cmd.sh  upgrade_dbs`

Note:

- This action will upgrade all databases to the last version.
- This action can take a while.
- you can reach this functionality only when Proline is set up (see how to setup Proline) .

Installing and configuring the Sequence Repository

Although this module is optional, it is recommended to install it to have access to the protein sequences in the user interfaces. The module can be installed from the Windows installer or from the archive file by simply extracting the content of the archive.

This module can be installed on the same computer running Proline Server. However, as it will parse Fasta files to extract sequence and description from them, the Sequence Repository will may be more efficient if it is installed on the computer where the Fasta files are stored. In any case, you must also be able to access the PostgreSQL server from this computer.

Sequence Repository Configuration

Configuration files are located in the folder `<seqrepo_folder>/config`.

Server and Datastore description

`application.conf` file defines the datastore and server description to access to the UDS database. Properties specified here should be the same as the one you specify while configuring the Proline Server.

```
proline-config {
 driver-type = "postgresql" // valid values are: h2, postgresql
 max-pool-connection=3  //Beta  properties  :  specify  maximum  number  of  pool
connected to DB Server. default to 3
}

//User and Password to connect to databases server.
auth-config {
 user="<user-proline>"
 password="<password-proline>"
}

//Databases server Host
host-config {
 host="<host>"
 port="5432"
}

uds-db {
 connection-properties {
 dbName = "uds_db" }
}
```

```
h2-config {
 script-directory = "/h2"
 connection-properties {
        connectionMode = "FILE"
        driver = "org.h2.Driver"
 }
}

postgresql-config {
 script-directory = "/postgresql"
 connection-properties {
        connectionMode = "HOST"
        driver = "org.postgresql.Driver"
 }
}
```

note:

- `auth-config:` Specify the PostgreSQL Proline user (and its password). They are the same as specified in `application.conf` for Proline Server

- `host-config:` IP address or name of the PostgreSQL server. Do not change anything else.

Protein description parsing rule

As this module is used to extract Protein sequence and description from a Fasta file for a specific protein accession, it is necessary to configure the rule used to parse the protein ACC, from Fasta description line. This is similar to the rules specified in Mascot Server, but these specified rules don't use the same semantic ! To do this, `parsing-rules.conf` file should be edited. In this file it is necessary to escape (this means prefix with '\') some characters: '\' , ':' and '='

```
//Specify path to fasta files for SeqRepository daemon. Multiple path separated
by ',' between []
//On linux system : local-fasta-directories =["/local/mascot/sequence"]
local-fasta-directories =["D:\\mascot\\sequence"]


// Rules used for parsing fasta entries. Multiple rules could be specified.
// name : identifying rule definition
// fasta-name : FASTA file name must match specified Java Regex CASE_INSENSITIVE.
multiple Regex separated by ',' between [] could be specified
// fasta-version : Java Regex with capturing group for fasta release version
extraction (CASE_INSENSITIVE)
// protein-accession : Java Regex with capturing group for protein accession
extraction


parsing-rules = [{
 name="label1",
 fasta-name=["uniprot"],
 fasta-version="[.]*_([^_]*).fasta",
 protein-accession =">\\w{2}\\|([^\\|]+)\\|"
},
{
 name="label2",
 fasta-name=["myDB"],
 fasta-version="[.]*_([^_]*).fasta",
 protein-accession =">\\w{2}\\|[^\\|]*\\|(\\S+)"
}
]


//Default Java Regex with capturing group for protein accession if fasta file
name doesn't match parsing_rules RegEx
// >(\\S+) : String after '>' and before first space
default-protein-accession =">(\\S+)"
```

For example, label1 rule will capture P07259 from line sp|P07259|PYR1_YEAST ... ⇒ a '>' then 2 char then '|' then (capture until '|') and '|'... This rule will be applied for all Fasta files prefixed with uniprot.

Testing rules

In order to verify the specified configuration, once previous files have been configured and saved, run the following tool under sequence repository installation directory `run-TestConfiguration.bat` (Windows) or `run-TestConfiguration.sh` (Linux).

An output will be displayed with all Fasta files found and for each which rule will be applied. The first 3 entries of each will also be displayed with the extracted protein accession.

Output should look like:

```
 Scanning [D:\temp\fasta]
 [D:\temp\fasta] scan terminated
 Number of traversed dirs: 1
 Found FASTA file names: 7
 ---- Scanning Fasta local path ----
 Using default rule ">(\S+)" for fasta "Nouvelle_base_données_sara.fasta"
 Accession "tr|F8WIX8|H2A.1_MOUSE" will be used for entry ">tr|F8WIX8|H2A.1_MOUSE
Original_Name=F8WIX8_MOUSE Histone H2A OS=Mus musculus GN=Hist1h2al PE=3 SV=1".
 Accession "tr|Q5M8Q2|H2A.L.1.3_MOUSE" will be used for entry
">tr|Q5M8Q2|H2A.L.1.3_MOUSE Original_Name= Q5M8Q2_MOUSE Histone H2A OS=Mus
musculus GN=OTTMUSG00000016789 PE=2 SV=1".
 Accession "tr|J3QP08|H2A.L.1.4_MOUSE" will be used for entry
">tr|J3QP08|H2A.L.1.4_MOUSE Original_Name= J3QP08_MOUSE Histone H2A OS=Mus
musculus GN=Gm14501 PE=3 SV=1".


[UPS1UPS2_D_20121108.fasta] matches Fasta Name Regex "UPS1UPS2_"
 Using rule ">[^\|]*\|(\S+)" for "UPS1UPS2_D_20121108.fasta"
 Release (using rule "_(?:D|(?:Decoy))_(.*)\.fasta") = "20121108"
 Accession "ALBU_HUMAN_UPS" will be used for entry ">P02768ups|ALBU_HUMAN_UPS
Serum albumin (Chain 26-609) - Homo sapiens (Human)".
 Accession "NEDD8_HUMAN_UPS" will be used for entry ">Q15843ups|NEDD8_HUMAN_UPS
NEDD8 (Chain 1-81) - Homo sapiens (Human)".
 Accession "RASH_HUMAN_UPS" will be used for entry ">P01112ups|RASH_HUMAN_UPS
GTPase HRas (Chain 1-189) - Homo sapiens (Human)".


 Using default rule ">(\S+)" for fasta
"uniprot-taxonomy%3A-Mus+musculus+%28Mouse%29+%5B10090%5D-.fasta"
 Accession "sp|Q9CQV8|1433B_MOUSE" will be used for entry ">sp|Q9CQV8|1433B_MOUSE
14-3-3 protein beta/alpha OS=Mus musculus GN=Ywhab PE=1 SV=3".
```

```
 Accession "sp|P62259|1433E_MOUSE" will be used for entry ">sp|P62259|1433E_MOUSE
14-3-3 protein epsilon OS=Mus musculus GN=Ywhae PE=1 SV=1".
 Accession "sp|P68510|1433F_MOUSE" will be used for entry ">sp|P68510|1433F_MOUSE
14-3-3 protein eta OS=Mus musculus GN=Ywhah PE=1 SV=2".


[iSa_D_20130403.fasta] matches Fasta Name Regex "ISA_"
 Using rule ">\w{2}\|([^\|]+)\|" for "iSa_D_20130403.fasta"
 Release (using rule "_(?:D|(?:Decoy))_(.*)\.fasta") = "20130403"
 Accession "Q99M51tag" will be used for entry ">sp|Q99M51tag|NCK1_strep-tag
Cytoplasmic protein NCK1 OS=Mus musculus GN=Nck1 PE=1 SV=1".
 Accession "Q9ES52tag" will be used for entry ">sp|Q9ES52tag|SHIP1_strep-tag
Phosphatidylinositol 3,4,5-trisphosphate 5-phosphatase 1 OS=Mus musculus
GN=Inpp5d PE=1 SV=2".
 Accession "###REV###Q99M51tag" will be used for entry
">sp|###REV###Q99M51tag|NCK1_strep-tag Reverse sequence, was Cytoplasmic protein
NCK1 OS=Mus musculus GN=Nck1 PE=1 SV=1".


[UP_MouseEDyP_D_20150629.fasta] matches Fasta Name Regex "UP_"
 Using rule ">\w{2}\|([^\|]*)\|\S+" for "UP_MouseEDyP_D_20150629.fasta"
 Release (using rule "_([^_])*\.fasta") = "9"
 Accession "Q9CQV8" will be used for entry ">sp|Q9CQV8|1433B_MOUSE 14-3-3 protein
beta/alpha OS=Mus musculus GN=Ywhab PE=1 SV=3".
 Accession "Q9CQV8-2" will be used for entry ">sp|Q9CQV8-2|1433B_MOUSE Isoform
Short of 14-3-3 protein beta/alpha OS=Mus musculus GN=Ywhab".
 Accession "P62259" will be used for entry ">sp|P62259|1433E_MOUSE 14-3-3 protein
epsilon OS=Mus musculus GN=Ywhae PE=1 SV=1".


[S_cerevisiae_Decoy_20121108.fasta] matches Fasta Name Regex "S_cerevisiae_"
 Using rule ">\w{2}\|([^\|]*)\|\S+" for "S_cerevisiae_Decoy_20121108.fasta"
 Release (using rule "_([^_])*\.fasta") = "8"
 Accession "P38903" will be used for entry ">sp|P38903|2A5D_YEAST
Serine/threonine-protein phosphatase 2A 56 kDa regulatory subunit delta isoform
OS=Saccharomyces cerevisiae (strain ATCC 204508 / S288c) GN=RTS1 PE=1 SV=2".
 Accession "P31383" will be used for entry ">sp|P31383|2AAA_YEAST Protein
phosphatase PP2A regulatory subunit A OS=Saccharomyces cerevisiae (strain ATCC
204508 / S288c) GN=TPD3 PE=1 SV=3".
 Accession "Q00362" will be used for entry ">sp|Q00362|2ABA_YEAST Protein
phosphatase PP2A regulatory subunit B OS=Saccharomyces cerevisiae (strain ATCC
204508 / S288c) GN=CDC55 PE=1 SV=2".


[UPS1UPS2_Decoy_20121108.fasta] matches Fasta Name Regex "UPS1UPS2_"
 Using rule ">[^\|]*\|(\S+)" for "UPS1UPS2_Decoy_20121108.fasta"
 Release (using rule "_(?:D|(?:Decoy))_(.*)\.fasta") = "20121108"
 Accession "ALBU_HUMAN_UPS" will be used for entry ">P02768ups|ALBU_HUMAN_UPS
Serum albumin (Chain 26-609) - Homo sapiens (Human)".
 Accession "NEDD8_HUMAN_UPS" will be used for entry ">Q15843ups|NEDD8_HUMAN_UPS
NEDD8 (Chain 1-81) - Homo sapiens (Human)".
 Accession "RASH_HUMAN_UPS" will be used for entry ">P01112ups|RASH_HUMAN_UPS
GTPase HRas (Chain 1-189) - Homo sapiens (Human)".
```

Running Sequence Repository

Sequence Repository can be run in two modes: as a daemon or as service on demand. In the first case, it will automatically scan for new protein sequences in every projects, with a given frequency (default is every two hours). In the last case, new sequences will only be retrieved when the user requests it for a specified identification summary. This feature is available in Proline Studio with the "Retrieve protein sequences" command.

Running as Service Daemon

To run Sequence Repository as a Daemon, you must modify and execute `run-RetrieveService.bat` (Windows) or `run-RetrieveService.sh` (Linux).

By default, RetrieveService will be run with option "-t 2" : the scan will be executed every 2 hours. Edit the run-RetrieveService script to change this value.

Other available options are:

```
Usage: <main class> [options]

Options:
 -f, --forceUpdate
 force update of MSIdb result summaries and biosequences (even if already
 updated)
 Default: false
 -p, --project
 the ID of the single project to process
 Default: 0
 -t, --time
 the daemon periodicity (in hours)
 Default: -1
 -debug
 set logger level to DEBUG (verbose mode)
 Default: false
```

Running as Service On Demand

To run Sequence Repository on demand, you must execute `run-RetrieveOnDemand.bat` (Windows) or `run-RetrieveOnDemand.sh` (Linux)
Note: the scripts could be modified to increase the amount of memory used.

Installation and configuration of Proline Web eXtension

The automated installer embeds several components of the Proline Suite. You may only install the Proline Web component, but if you want you can perform a full installation. It will install Proline-Admin and Proline-WebCore in the bin directory of Proline Web.

To begin the installation download and execute the PWX automated installer.

Configure the PWX Server

The Proline Web eXtension (PWX) is based on the use of MongoDB database engine. By default, the MongoDB engine is installed by the automated installer. However, if you installed your MongoDB database on a different computer than the PWX server, you'll need to edit the PWX configuration file:

- Go to the installation directory of your PWX server
- Go to the “conf” folder and open the “application.conf” file with any text editor (like Window default Notepad)
- Edit the mongodb.servers and cache.mongodb.servers parameters by setting the host name and the port number corresponding to your MongoDB server (default MongoDB port is 27017)

If MongoDB is installed on the same computer as the PWX server, you don't need to configure anything at this time.

Setup the Proline Server connection

The connection between PWX and the Proline server is required to execute the Proline computational tasks. By default PWX will try to connect to the Proline on the same computer. If you want to deploy PWX on a different computer then the connection needs to be configured. To do this, open the file named “application.conf” located in the “conf” folder go to the Proline section of the file. Then edit the jms-server “host” and type the name of computer where the Proline server is installed.

You will need to restart the PWX server to make sure these new settings are updated.

Change the Postgres database connection identifiers

By default, PWX will connect to PostgreSQL database using the default connection identifiers (postgres/postgres). If you want to use a different user name or different passwod then you have to change it in the PWX configuration file. To do this, open the file named “application.conf” located in the “conf” folder go to the Proline section of the file. Then edit the auth-config properties to update your connection identifiers.

You will need to restart the PWX server to make sure these new settings are updated.

Launch the PWX Server

- First, please make sure that MongoDB is running. If it is not, please start it manually. See official docs
- In order to start the PWX server, go to its installation folder. On Windows platforms launch the “start.bat” script. On Linux platforms execute the “start.sh” script.

Connect to the Proline Desktop

Once PWX is running, you can connect to the Proline Web eXtension by opening a Web Browser and go to the address of this form: “name-of-the-machine:9000” or “local.ip.of.the.machine:9000” (for instance 192.168.0.30:9000). The default user is “admin” and its password is “proline”. Don't forget to change its password from the User menu button once you're logged in.