



*Inventory, Network  
discovery & Provisioning  
3.0  
User & Administration  
Guide*



**Jilroy Software LTD 2010**

Jilroy Software makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Jilroy Software shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Jilroy Software. The information contained in this document is subject to change without notice.

Copyright © 2000-2010 Jilroy Software. All rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

Microsoft® is a US registered trademark of Microsoft Corporation.

Windows® and MS-Windows® are US registered trademarks of Microsoft Corporation.

Pentium® is a US registered trademark of Intel Corporation.

UNIX® is a registered trademark of The Open Group.

All other company and product names may be trademarks or registered trademarks of their respective owners.

## Table of Contents

Preface.....	6
Introduction.....	8
An Overall view of the product.....	8
The Inventory, Network Discovery & Provisioning command lines.....	9
Utility commands.....	9
Scheduler.....	9
The application server (tomcat).....	9
GUI.....	9
External Report Generator.....	10
The Inventory, Network Discovery & Provisioning product functionality.....	10
Advantages.....	10
Getting Started on Unix Platforms.....	12
Unix System Requirements.....	12
Installation on UNIX.....	12
Getting the software.....	12
Installing Inventory, Network Discovery & Provisioning 3.0.....	12
License.....	14
Execution Permission.....	14
Post Installation scripts on UNIX.....	14
Post-Installation scripts.....	14
Database configuration.....	15
Automatic Startup.....	15
Un-installation on UNIX.....	16
removing all presence of the product.....	16
Remove only Inventory, Network Discovery & Provisioning binaries.....	16
remove the crontab additions.....	16
GUI only Installation on Unix Platforms.....	17
Unix System Requirements.....	17
Installation on UNIX.....	17
Getting the software.....	17
Installing the GUI.....	17
License.....	18
Execution Permission.....	18
Post Installation scripts on UNIX.....	18
Un-installation on UNIX.....	19
removing all presence of the product.....	19
Getting Started on Windows Platforms.....	20
Windows System Requirements.....	20
Installation on Windows.....	20
Getting the software.....	20
Installing Inventory, Network Discovery & Provisioning 3.0.....	20

License.....	25
Database configuration.....	25
Automatic Startup.....	25
Un-installation on Windows.....	26
GUI only Installation on Windows Platforms.....	28
Windows System Requirements.....	28
Installation on Windows.....	28
Getting the software.....	28
Installing Inventory, Network Discovery & Provisioning 3.0 GUI.....	28
License.....	32
Un-installation on Windows.....	33
Configuration.....	35
Config.prop.....	35
General.....	36
Database.....	36
Discovery.....	37
snmp_communities.....	37
DiscoveryJobs.....	37
config.xml.....	38
Changing the Database scheme.....	38
configGUI.prop.....	40
configGUI.xml.....	40
provisioning.csv.....	42
IpIncludeListFile.txt.....	44
IpExcludeListFile.txt.....	45
The Discovery & Provision process logic.....	46
The discovery process.....	46
The current discovery table scheme.....	47
tblObject.....	47
tblArp.....	49
tblVlanInfo.....	49
tblObjectLastMonitoringTime.....	50
Discovery Methods.....	50
The Discovery command lines.....	51
discovernet.....	51
buildprovisioningfromarp.....	52
discovernetworkwitharp.....	52
inventory.....	53
Utility commands.....	54
snmptrap.....	54
The Inventory, Network Discovery & Provisioning GUI.....	55
The product menu.....	55
The Product's Panels.....	57
Edit Product Config File.....	57
Edit Provisioning File.....	58
Edit GUI Config File.....	59

License Info.....	60
Run Discovery commands.....	60
Discovery Jobs Status.....	61
Nodes Report.....	63
Interfaces Report.....	64
Vlans Report.....	66
ARP Table Report.....	68
About.....	69
Backup and Restore for the Database.....	70
exporttable2csv.....	70
importcsv2table.....	70
Additional SQL related commands.....	72
Query analyzer for the database.....	72
createdb.....	72
createtables.....	72
dbconf.....	72
loadtableinfo.....	73
sqlexec.....	73
startdb.....	73
stopdb.....	73
truncatetables.....	74
The product Scheduler.....	75
The Scheduler's tables structure.....	76
tblJobs.....	76
configDbScheduler.xml.....	76
Categories file.....	77
GUI.....	78
Internal commands.....	79
Loading jobs to the database.....	80
The product Security mechanism.....	82
The Users and Permissions table.....	82
tblUsers.....	82
tblPermissions.....	83
The connection.prop file.....	83
The Security GUI.....	85
General Commands permitted.....	89
Final Page.....	90

# Preface

---

Welcome to Inventory, Network Discovery & Provisioning 3.0. This chapter provides an introduction to the structure and assumptions of this guide.

## The Purpose of This Guide

This guide contains information needed to use Inventory, Network Discovery & Provisioning 3.0 from Jilroy Software efficiently and effectively.

## Who Should Use This Guide

This guide is intended for System & network managers familiar with system & network management and its fundamental concepts.

## Organization of This Guide

This guide is structured to reflect the following conceptual divisions:

- Preface – A description of the guide's purpose, intended audience, organization, and conventions.
- Introduction section – A general description of Inventory, Network Discovery & Provisioning 3.0 and its goals, its way of operation and general guidelines
- Installation section – Information about installing, un-installing, and configuring Inventory, Network Discovery & Provisioning 3.0.
- The product's User Interface section – Contains information about the specific User Interface of the product.
- Additional General capabilities and services section – Contains information about general features like the Scheduling mechanism, and the security implementation.
- Final Page – Information about contacting Jilroy Software.

## Conventions

The manual uses the following conventions:

- Names of dialog boxes, windows, and unnamed screen areas are displayed in *italics*.
- Names of buttons, tabs, check-boxes, and other screen elements are displayed in **bold**. For example, click **OK** or type the **Start date**.
- **This font** is used for text that you enter.
- `This font` is used for code, directory names, file names, and system activity.
- UPPERCASE is used for keys and acronyms.
- Steps that involve two or more selections from a menu may be presented as a combination of selections separated by an > angled bracket.

For example, when you see **File > New**, click the **File** menu on the menu bar. This will open a drop-down menu. Then select the **New** command.

- Cross-references are underlined. For example, see Chapter 2.
- Hyperlinks are underlined and [blue](#).
- The ⓘ symbol signifies notes, which are used to provide extra or special information regarding the preceding topic.
- The *Italic* font style is used to *emphasize* words and phrases in special cases.

## Introduction

---

Inventory, Network Discovery & Provisioning 3.0 from Jilroy Software is a full scale System & network discovery product which enables you to learn inventory information about your organization in an easy way, and access it using its own reports or directly using SQL. It also enables you to perform automated actions during and based on the discovery based inventory process.

### *An Overall view of the product*

The Inventory, Network Discovery & Provisioning product is composed of multiple components, that together create its entire functionality. In the following sections we will describe the functionality of each component.



**i** Note: The Inventory, Network Discovery & Provisioning Core is built on top of the Generic discovery Engine, which can be used for performing any discovery operation, and not only network discovery.

The Inventory, Network Discovery & Provisioning application is just a sample of the Discovery Engine capabilities.



Please look at the Discovery Engine documentations for extending the discovery capabilities of the product.

## **The Inventory, Network Discovery & Provisioning command lines**

The actual discovery operation is performed using the inventory command lines. These are a set of batch commands, that are doing the discovery & provisioning operations based on discovery rules defined to the product. Please look at the generic Discovery engine documentation on how to extend the discovery schema and operation.

Those commands can be executed and scheduled from the supplied User Interface.

## **Utility commands**

The product comes with a set of utility commands that come to answer all user's needs regarding the day to day operation of the product.

## **Scheduler**

The product comes with a built in scheduler which runs periodically the discovery operations (if needed) and runs event handlers in case of node additions and deletions.

## **The application server (tomcat)**

The product comes with an application server which handles users requests coming from the GUI of the product.

## **GUI**

The product comes with a Java GUI which enables the user to manage the product, and see reports about the discovered network.

## External Report Generator

We added a free-ware report generator, although we supply reporting tools within the product gui, for the user's convenience. The report generator is named: datavision, you can find its documentation in its install directory

## *The Inventory, Network Discovery & Provisioning product functionality*

The Inventory, Network Discovery & Provisioning product can have multiple purposes:

1. It can be a part of a CMDB that performs a discovery for the enterprise.
2. Act as the core of a monitoring platform, supplying inventory information about the monitored network.
3. Acts as a core of a provisioning system.
4. and much more...

## *Advantages*

- It is a full Java based solution and as such can run on any java supporting platform from Unix platforms (Linux/sun/hpux) to Windows and others.
- You can alter and customize the collected fields on the discovered nodes using configuration files, so now you can tailor the discovery process to your exact needs.
- You can decide whether the product will auto discover the nodes in your network or you can specify directly which devices will be discovered
- Its output is redirected into a database which can be accessed using jdbc

Jilroy Software encourages you to use the product and extend it.

If you have any questions and recommendations about how to use or how to improve the product you are most welcome to send us your thoughts to [info@jilroy.com](mailto:info@jilroy.com)

If you need us to tailor the product to your exact needs we are offering this kind of service for reasonable fees.

Jilroy Software LTD

Inventory, Network Discovery & Provisioning

## Getting Started on Unix Platforms

---

This chapter contains information about prerequisites, installing and uninstalling the product on UNIX operating systems.

### *Unix System Requirements*

Inventory, Network Discovery & Provisioning 3.0 is a full JAVA based solution from the application to the database. It comes with a JRE and an Instance of a database. So it comes equipped with anything needed for its run, and it does not conflict with any existing JREs on your machine.

### *Installation on UNIX*

This section contains information on how to get the installation package and on how to install it on UNIX operating systems.

#### **Getting the software**

To get the installation package, you can get it from the Jilroy web site [www.jilroy.com](http://www.jilroy.com), from the download section.

If you have a download ready for your type of machine, you do not need to prepare anything.

#### **Installing Inventory, Network Discovery & Provisioning 3.0**

The product comes compressed in a tar.gz format.

Use the following commands to open the downloaded file.

1. **gunzip filename.tar.gz**  
This will extract the compressed installation file, into the tar format.
2. **tar -xvf filename.tar**  
This will extract all the product files to the /opt/jilroy directory

**i** By default, Inventory, Network Discovery & Provisioning 3.0 installs itself to the /opt/jilroy directory.

To Install to a different file system, use the **ln -s** command to create a soft-link from /opt/jilroy to the new destination directory.

**i** In case of re-install of the software do networkdiscovery\_stop prior to the uninstall process.

The package installation phase creates the directory structure shown in Table 3.0

Table 2.1:Unix Directory structure

Directory	Description
/opt/jilroy	main directory
/opt/jilroy/jre	jre for current operating system
/opt/jilroy/hsqldb	hsqldb database directory
/opt/jilroy/hsqldb/database	the location of the databases created. it will be empty on installation
/opt/jilroy/hsqldb/database/rjnetworkdiscovery	the directory that will contain the database of the Inventory, Network Discovery & Provisioning program.
/opt/jilroy/rjnetworkdiscovery	the home directory for the Inventory, Network Discovery & Provisioning application.
/opt/jilroy/rjnetworkdiscovery/bin	the bin directory of the application
/opt/jilroy/rjnetworkdiscovery/conf	the configuration files directory
/opt/jilroy/rjnetworkdiscovery/conf/discoveryrules	the discovery rules configuration files location
/opt/jilroy/rjnetworkdiscovery/lib	jar files used by the program
/opt/jilroy/rjnetworkdiscovery/lic	license information for open source software used
/opt/jilroy/rjnetworkdiscovery/log	logs directory
/opt/jilroy/rjnetworkdiscovery/temp	temporary files dir

## License

The product comes with trial licenses for you to be able to test the product. For permanent licenses please contact us at [sales@jilroy.com](mailto:sales@jilroy.com).

The License files should be located in the `/opt/jilroy/rjnetworkdiscovery/conf` directory.

## Execution Permission

The Inventory, Network Discovery & Provisioning product does not require to run under root permission, however if you want to use ICMP based discovery in the product, you will need root permissions to run this protocol.

## Post Installation scripts on UNIX

To complete the installation, a script must be executed. This script will create the database of the product and the tables needed for its operation.

### Post-Installation scripts

The post-installation script performs the following actions:

1. Start the database server.
2. Create the database needed for the application.
3. Create the tables needed for the application.
4. Fills system tables with default values

The script can be found in the binaries directory:

`/opt/jilroy/rjnetworkdiscovery/bin`

to execute the script just enter its name **`./install`** from the *bin* directory and press ENTER.

**i** The 'install' script launches some services that should be entered into the cron tab. The services are: The hsql Database, the tomcat application server, and the scheduler. Please remember to enter them into the cron tab


## Database configuration

The default database used by the program is hsql-db which is an open source project by the Apache team. It is a full JAVA database.

The database is installed with the product.

The connection information of the database can be found in:  
`/opt/jilroy/rjnetworkdiscovery/conf/config.prop`

By default the connection information will work with the supplied installation file.

 It is possible to make the program run with different database such as Oracle or PostgreSQL. The only requirement is that the database will have a JDBC driver. Look later in the book on how to change the database used by the product.

## Automatic Startup

The ***install*** script starts the database. However in order to start the database after a boot to the system, add to your scheduler (crontab) the following commands:

```
/opt/jilroy/rjnetworkdiscovery/bin/startdb
```

This will start the database server. Currently the port on which the database is running is an internal port with the number of 54345. This can be changed by the user. Look on later chapters on how to change the database listening port if needed.

```
/opt/jilroy/rjnetworkdiscovery/bin/runscheduler
```

This will launch the product scheduler. It runs the commands launched by the GUI, and the event handlers.

```
/opt/jilroy/rjnetworkdiscovery/bin/runtomcat.sh
```

This command launches the application server used by the GUI.

## Un-installation on UNIX

The product has only presence on the /opt/jilroy directory.

### **removing all presence of the product**

If you have only installed the Inventory, Network Discovery & Provisioning 3.0 product, and want to remove all its presence, then remove the jilroy directory using the command:

```
rm -r /opt/jilroy
```

This will remove all the installed parts of the product

### **Remove only Inventory, Network Discovery & Provisioning binaries**

If you have installed more products from jilroy Software, then they are using the jre, and the hsqldb directories. In this case you must not delete those directories.

In order to delete only the Inventory, Network Discovery & Provisioning binaries do:

```
rm -r /opt/jilroy/rjnetworkdiscovery
```

in order to delete the Inventory, Network Discovery & Provisioning database too, then do:

```
rm -r /opt/jilroy/hsqldb/database/rjnetworkdiscovery
```

### **remove the crontab additions**

Remember to remove the commands related to the product which you entered into the crontab. Look at the Automatic startup section for the recommended crontab commands.



## GUI only Installation on Unix Platforms

---

This chapter contains information about the actions needed in order to install only GUI of the product on a different machine from the Inventory, Network Discovery & Provisioning engine.

### *Unix System Requirements*

The GUI is a full JAVA application which works as a stand-alone application, and connects remotely to the Network-Discovery application server. It comes with a JRE so it does not conflict with any existing JREs on your machine.

### *Installation on UNIX*

This section contains information on how to get the installation package and on how to install it on UNIX operating systems.

#### **Getting the software**

To get the installation package, you can get it from the Jilroy web site [www.jilroy.com](http://www.jilroy.com), from the download section.

If you have a download ready for your type of machine, you do not need to prepare anything.

#### **Installing the GUI**

The product comes compressed in a tar.gz format.

Use the following commands to open the downloaded file.

3. **gunzip filename.tar.gz**  
This will extract the compressed installation file, into the tar format.
4. **tar -xvf filename.tar**  
This will extract all the product files to the /opt/jilroy directory

❗ By default, Inventory, Network Discovery & Provisioning GUI installs itself to the /opt/jilroy directory.

To Install to a different file system, use the **ln -s** command to create a soft-link from /opt/jilroy to the new destination directory.

If you have already an Inventory, Network Discovery & Provisioning server installed. Do NOT install the GUI too. It will override some of the server files

The package installation phase creates the directory structure shown in Table 2.1

Table 2.1:Unix Directory structure

Directory	Description
/opt/jilroy	main directory
/opt/jilroy/jre	jre for current operating system
/opt/jilroy/rjnetworkdiscovery	the home directory for the Inventory, Network Discovery & Provisioning GUI application.
/opt/jilroy/rjnetworkdiscovery/bin	the bin directory of the application
/opt/jilroy/rjnetworkdiscovery/conf	the configuration files directory
/opt/jilroy/rjnetworkdiscovery/lib	jar files used by the program
/opt/jilroy/rjnetworkdiscovery/lic	license information for open source software used
/opt/jilroy/rjnetworkdiscovery/log	logs directory
/opt/jilroy/rjnetworkdiscovery/temp	temporary files dir

## License

The GUI does not need a license file.

## Execution Permission

The Inventory, Network Discovery & Provisioning GUI does not require to run under root permission.

## Post Installation scripts on UNIX

After the opening of the tar file, the only thing needed is to launch the GUI using the networkdiscoverygui.sh script.

By default the GUI tries to find the Inventory, Network Discovery & Provisioning Application server on the local machine. You have to change the ServerURL configuration parameter and point to the address of the right sever. This is done from the main menu at: Administration -> Edit GUI Config File

See the chapter on the GUI for detailed explanation on how to do this.

## Un-installation on UNIX

The product has only presence on the /opt/jilroy directory.

### **removing all presence of the product**

If you have only installed the Inventory, Network Discovery & Provisioning 3.0 GUI, and want to remove all its presence, then remove the jilroy directory using the command:

```
rm -r /opt/jilroy
```

This will remove all the installed parts of the product

## Getting Started on Windows Platforms

---

This chapter contains information about prerequisites, installing and uninstalling the product on the windows operating systems.

### *Windows System Requirements*

Inventory, Network Discovery & Provisioning 3.0 is a full JAVA based solution from the application to the database. It comes with a JRE and an Instance of a hsql database. So it comes equipped with anything needed for its run, and it does not conflict with any existing JREs on your machine.

### *Installation on Windows*

This section contains information on how to get the installation package and on how to install it on a Windows operating systems.

### **Getting the software**

To get the installation package, you can get it from the Jilroy web site [www.jilroy.com](http://www.jilroy.com), from the download section.

If you have a download ready for your type of machine, you do not need to prepare anything.

### **Installing Inventory, Network Discovery & Provisioning 3.0**

The product comes with an executable program that contains the installation product. The product is called: JilroyNetworkDiscovery.vxy.yyymmdd.exe.

Where

Vxy – stands for version x, sub-version y.

yyymmdd - is the date of generation of the installation.

You have to run the installation executable, and the installation program will guide you through.

**i** The installation process asks you to select an installation path for the product. You must not select a path that contains spaces in the directory names. If you do, there might be problems when running the product.

The package installation phase creates the directory structure shown in Table 3.1

Table 3.1:Windows Directory structure

Directory	Description
[INSTALLDIR]\jilroy	main directory
[INSTALLDIR]\jre	jre for current operating system
[INSTALLDIR]\hsqldb	hsql database directory
[INSTALLDIR]\hsqldb\database	the location of the databases created. it will be empty on installation
[INSTALLDIR]\hsqldb\database\rjnetworkdiscovery	the directory that will contain the database of the Inventory, Network Discovery & Provisioning program.
[INSTALLDIR]\rjnetworkdiscovery	the home directory for the Inventory, Network Discovery & Provisioning application.
[INSTALLDIR]\rjnetworkdiscovery\bin	the bin directory of the application
[INSTALLDIR]\rjnetworkdiscovery\conf	the configuration files directory
[INSTALLDIR]\rjnetworkdiscovery\conf\discoveryrules	the discovery rules configuration files location
[INSTALLDIR]\rjnetworkdiscovery\lib	jar files used by the program
[INSTALLDIR]\rjnetworkdiscovery\lic	license information for open source software used
[INSTALLDIR]\rjnetworkdiscovery\log	logs directory
[INSTALLDIR]\rjnetworkdiscovery\temp	temporary files dir

***The installation process***

After double clicking the installation executable you will see the following screen.

This is the first screen of the install process. Press Next to continue.

At this stage the following screen will appear:

This is the license agreement document. Please read it carefully and only if you

agree then type I agree to continue.

Now you are requested to select the installation path.

**i** You must NOT select a path that contains spaces in the directory names. If you do, there might be problems when running the product.

Now press Install to start the actual install of the product. During the installation process the following screen will appear.

The pop-up window instructs you to copy the license files you got from Jilroy Software to the [INSTALLDIR]\conf directory, if you have them, and if not you can still use the product with its trial license which time limited. The time limit is for 30-45 days from the download date of the installation executable.

During the install process the following command window will appear.

This happens as the install process does a few actions during the install process. The actions are:

1. Install 3 services that are used by the product:
  - JilroyDbServer: This is a service that runs the database used by the product. The database is a known open-source java database called hsql.
  - JilroyScheduler: This is a service that runs jobs scheduled by the user or by the

application it self.

- JilroyTomcat: This is the application server used to serve the GUI clients of the application.
2. Initiate the database scheme and create the tables used by the product
  3. Register the application GUI in the computer's start-up programs.

At the end of the install process the following screen appears:

Press Finish to end the installation.



❗ You can look at the install script in the [INSTALLDIR]\bin\install.bat.

## License

The product comes with trail licenses for you to be able to test the product. They usually last for 30-45 days from the download time of the install program. For permanent licenses please contact us at [sales@jilroy.com](mailto:sales@jilroy.com).

The License files should be located in the [INSTALLDIR]\rnetworkdiscovery\conf directory.

## Database configuration

The default database used by the program is hsql-db which is an open source project by the Apache team. It is a full JAVA database.

The database is installed with the product.

The connection information of the database can be found in:  
[INSTALLDIR]\rjnetworkdiscovery\conf\config.prop

By default the connection information will work with the supplied installation file.

❗ It is possible to make the program run with different database such as Oracle or PostgreSQL. The only requirement is that the database will have a JDBC driver. Look later in the book on how to change the database used by the product.

## Automatic Startup

The **install** script defines the services used by the product. Make sure not to change that they will start automatically when the computer starts.

## Un-installation on Windows

The product has multiple methods for un-installing it. The best way is to use the Add/Remove programs from the control panel.

The following screens will appear.

Press Next to continue.


The following screen will appear. It tells you what you are about to do:

Press the Uninstall button.

During the un-install process the following screen will appear:

The un-install will remove the defined services and the defined launch commands.  
In the end of the un-install program the following screen will appear:

Press Finish to end the install process.

 The product keeps a backup of its conf directory in the %TEMP%/jilroy. This is done as the user may add functionality to the product that might be important to him.

## **GUI only Installation on Windows Platforms**

---

This chapter contains information about the actions needed in order to install only the product's GUI on a machine different from the Inventory, Network Discovery & Provisioning engine.

### *Windows System Requirements*

The GUI is a full JAVA application which works as a stand-alone application, and connects remotely to the Network-Discovery application server. It comes with a JRE so it does not conflict with any existing JREs on your machine.

### *Installation on Windows*

This section contains information on how to get the installation package and on how to install it on a Windows operating systems.

### **Getting the software**

To get the installation package, you can get it from the Jilroy web site [www.jilroy.com](http://www.jilroy.com), from the download section.

If you have a download ready for your type of machine, you do not need to prepare anything.

### **Installing Inventory, Network Discovery & Provisioning 3.0 GUI**

The product comes with an executable program that contains the installation product. The product is called: JilroyNetworkDiscoveryGUI.vxy.yyymmdd.exe.

Where

Vxy – stands for version x, sub-version y.

yyymmdd - is the date of generation of the installation.

You have to run the installation executable, and the installation program will guide you through.

**i** The installation process asks you to select an installation path for the product. You must not select a path that contains spaces in the directory names. If you do, there might be problems when running the product.

The package installation phase creates the directory structure shown in Table 3.1

Table 3.1:Windows Directory structure

Directory	Description
[INSTALLDIR]\jilroy	main directory
[INSTALLDIR]\jre	jre for current operating system
[INSTALLDIR]\rjnetworkdiscovery	the home directory for the Inventory, Network Discovery & Provisioning application.
[INSTALLDIR]\rjnetworkdiscovery\bin	the bin directory of the application
[INSTALLDIR]\rjnetworkdiscovery\conf	the configuration files directory
[INSTALLDIR]\rjnetworkdiscovery\lib	jar files used by the program
[INSTALLDIR]\rjnetworkdiscovery\lic	license information for open source software used
[INSTALLDIR]\rjnetworkdiscovery\log	logs directory
[INSTALLDIR]\rjnetworkdiscovery\temp	temporary files dir

The installation process

After double clicking the installation executable you will see the following screen.

This is the first screen of the install process. Press Next to continue.

At this stage the following screen will appear:

This is the license agreement document. Please read it carefully and only if you agree then type I agree to continue.

Now you are requested to select the installation path.


**i** You must NOT select a path that contains spaces in the directory names. If you do, there might be problems when running the product.

Now press Install to start the actual install of the product. During the installation process.

During the installation process, the GUI files are copied and there is a registration of the application GUI in the computer's start-up programs.

At the end of the install process the following screen appears:

Press Finish to end the installation.

 You can look at the install script in the [INSTALLDIR]\bin\install.bat.

## License

No license is needed for the GUI installation, so you may install any number of clients as needed. However the product itself requires a license.



## Un-installation on Windows

The product has multiple methods for un-installing it. The best way is to use the Add/Remove programs from the control panel.

The following screens will appear.

Press Next to continue.

The following screen will appear. It tells you what you are about to do:

Press the Uninstall button.

The un-install will remove the program files and the defined launch commands.  
In the end of the un-install program the following screen will appear:

Press Finish to end the install process.

# Configuration

---

In this chapter we will explain how to configure Inventory, Network Discovery & Provisioning 3.0 from the most common parts to the lesser common ones.

❗ In this chapter you will not find explanation on the configuration of general components (like the scheduler). The information about them will be found in their specific chapter.

## *Config.prop*

### **[General]**

Verbose=FINEST

full\_export\_freq=24:00:00

### **[Database]**

# derby definitions

#DriverName=org.apache.derby.jdbc.ClientDriver

#Manager=jdbc:derby://127.0.0.1:54345/rjnetworkdiscovery

#hsqldb definitions

DriverName=org.hsqldb.jdbcDriver

Manager=jdbc:hsqldb:hsql://127.0.0.1:54345/rjnetworkdiscovery

User=jilroy

Password=jilroy

### **[snmp\_communities]**

Community1=public

Community2=public2

Community3=public3

Community4=public4

```
Community5=  
Community6=  
[Discovery]  
launch_freq=24:00:00  
rediscovery_timeout=14:00:00:00  
[DiscoveryJobs]  
Add_Node_job=[INSTALLDIR]bin/printline.sh [INSTALLDIR]log/printline.log  
Delete_Node_job=[INSTALLDIR]bin/printline.sh [INSTALLDIR]log/printline.log
```


/conf/Config.prop is the user configuration file, for the batch commands of the discovery process. It contains the the following sections:

## General

This section defines general parameters regarding the operation of the software. Currently the parameters that can be set are:

### **Verbose**

Defines the detail level of the logs produced by the software. valid values are: severe , info, fine, finer and finest.

 If you want to change the Verbose value, you have to restart the relevant processes after the change as they read this value only on the 'init' phase. used by the product.

## Database

This section defines the connection string info for the discovery database. Out of the box, the system comes with the hsql Database, running in its server mode, on port 54345.

Currently the parameters that can be set are:

### **DriverName**

The Class name for the database client.

### **Manager**

The JDBC connection string of the database

**User**

The User Name used to access the database

**Password**

The Password value used to access the database

You can alter all these parameters to fit your environment.

**i** Note that currently the ./startdb and ./stopdb scripts are pointing to the hsql Db scripts. You will have to fix them too so they will fit your database.

## Discovery

Discovery related parameters.

**launch\_freq**

The program has an internal scheduler. This parameter specifies the frequency that the full discovery process should be automatically launched. A value of 00:00:00 means no launch at all.

**rediscovery\_timeout**

The timeout after which the discovery process will perform a new discovery for the existing node. The default is 14 days. A value of 0, means no operation performed.

## snmp\_communities

You can specify a list of up to 6 communities that will be tested for every node you want to discover. Write the communities you want near the :  
Community# =

The order in which they are written will be the order that they will be checked.

## DiscoveryJobs

The product supports an option to launch user defined jobs when certain discovery related events happen.

Currently there are two events supported:

- of You can specify a list of up to 6 communities that will be tested for every node you
- Add\_Node\_job – The command launched when a new node is added.
- Delete\_Node\_job – The command launched when a node is deleted.

The jobs receive as parameters the following values:

- The Monitoring Method
- The IP Address

## config.xml

The conf/config.xml file is an internal configuration file, that would hardly have to be processed by the user. However this is the main configuration file passed as a parameter to all the batch related command lines.

Please do not change the values in this configuration file unless you are specifically told by the Jilroy technicians, or you fully understand the meaning of the changes.

## Changing the Database scheme

One of the sub-directories of conf/ directory is named conf/tables. It contains files that have tables definitions. Those files define the database scheme. These definition files are used when the product is installed, and when the *dbconf* script is executed.

Currently only 2 types of fields are supported:

- String
- Int

The format of a definition file is:

```
<Tables>
  <Table>
    <Name>tblObject</Name>
    <Fields>
      nObjectId int,
```

```
        nParentObjectId int NULL,  
        ...  
    </Fields>  
</Table>  
    ...  
<Table>  
    ....  
</Table>  
<Tables>
```

**Tables**

A Wrapper tag that contains the tables definitions. It contains Table definitions

**Table**

```
<Table>  
    <Name>tblObject</Name>  
    <Fields>  
        nObjectId int,  
        nParentObjectId int NULL,  
        ...  
    </Fields>  
</Table>
```

The table definition XML contains the following tags:

**Name**

The name of the table

**Fields**

The Fields that will be in the specified table.

The product was tested on the formats of field definitions for the following

databases:

- hsql DB
- Derby Db
- SQL Server
- PostGreSQL
- Oracle

And is expected to work with any database supporting JDBC.

## configGUI.prop

```
[General]
Verbose=FINEST
ServerURL=http://127.0.0.1:54346/
```

//conf/ConfigGUI.prop is the user configuration file, for the Inventory, Network Discovery & Provisioning GUI. It contains the the following sections:

### General

This section defines general parameters regarding the operation of the software. Currently the parameters that can be set are:

#### **Verbose**

Defines the detail level of the logs produced by the software. valid values are: severe , info, fine, finer and finest.

❗ If you want to change the Verbose value, you have to restart the relevant processe after the change as they read this value only on the 'init' phase. used by the product.

#### **ServerURL**

This parameter points to the address of the Application server of the product. This value should not be changed if the GUI is launched on the server's machine.

## configGUI.xml

The conf/configGUI.xml file is an internal configuration file, that would hardly have



to be processed by the user. However this is the main configuration file passed as a parameter to the GUI

## provisioning.csv


The conf/provisioning.csv file contains instruction for the network discovery command line, as to what to discover. Refer to the “Discovery process logic” for a detailed description on the file usage.

```
#format: single line provisioning entries
#-----
#ip_addr,discovery_method,MonitoringMethod,ForceDiscovery,community,password,Snmpversion,MonitoringFrequencySec,TimeoutMsec,Retry,ServiceLevel,GroupingLevel1,GroupingLevel2,GroupingLevel3,GroupingLevel4
#
# where:
# discovery method=none|SNMP
# monitoring method=ICMP|SNMP (used by the monitoring platform)
# force discovery=0|1
# community acts as a userid for snmpv3
# password is only relevant for snmp v3
# snmp version=1|2|3
# Monitoring Frequency (used by the monitoring platform)
# ServiceLevel=(optional)Any Category of service level (Platinum, Gold, Silver,Bronze,...)
# GroupingLevel1 - GroupingLevel4= (optional) Grouping level
#           for example location   : 1- state, 2-city, 3-street, 4-building
#           or                     : company, state, city, department
10.0.1.201,ICMP,ICMP,0,none,none,1,300,3000,3,Gold,IBM,France,Paris,Marketing
10.0.1.202,ICMP,ICMP,0,none,none,1,200,3000,3,Gold,IBM,England,London,Finance
10.0.1.203,ICMP,ICMP,0,none,none,1,300,3000,3,Platinum,Jilroy,US,NY,Management
10.0.1.254,SNMP,SNMP,1,VOLVO,none,1,300,3000,3,Platinum,Jilroy,US,NY,RND
```

Remarks in this file are lines starting with the character '#'

The command lines have a strict format of a CSV that have the following ordered list of parameters:

1. IP address – The IP address of the node to discover.
2. Discovery Method – Currently there are 2 methods of discovery:
  - SNMP – The node should be discovered using the SNMP protocol.
  - ICMP – The node should only be added to the database, but no-detailed discovery is requested. This option is useful when working with the Monitoring Platform product.
3. Monitoring Method – The monitoring method that should be used for the product. This parameter is relevant when working with the Monitoring platform product. Currently the methods supported are:
  - SNMP – Monitoring the devices interfaces using SNMP
  - ICMP – Monitor the given node address using ICMP echo requests (Ping)
4. Force Discovery – This parameter specifies that the discovery program will perform a rediscovery for this node, even if it was already discovered, and its rediscovery timeout did not expire
5. Community/user – The community of the node for SNMP operation. If SNMP V3 is requested, then this value will specify the user value.
6. Password – The password value for SNMP V3 operations. It can contain the 2 password for authentication and privacy
7. SNMP version – the version of SNMP that the device supports.. Currently supported values are 1,2.
8. Monitoring Frequency in seconds – Used by the monitoring platform to define the frequency of performing a monitoring action
9. Timeout in milliseconds – The time we wait for reply from the device on a request
10. Retry count – The retry count on a request if no reply was supplied.
11. ServiceLevel – (optional parameter) Any Category of service level (Platinum, Gold, Silver, Bronze,...)
12. GroupingLevel1 – The 1<sup>st</sup> level of grouping for the node.
13. GroupingLevel2 – The 2<sup>nd</sup> level of grouping for the node.
14. GroupingLevel3 – The 3<sup>rd</sup> level of grouping for the node.
15. GroupingLevel4 – The 4<sup>th</sup> level of grouping for the node.

 The ServiceLevel and GroupingLevels are used for documentation reasons in the MonitoringPlatform product.

❗ Currently SNMP V3 is not supported. If this is needed please contact us.

## IpIncludeListFile.txt

The conf/IpIncludeListFile.txt file contains a list of IP masks that will be added to the discovery database. If no entry given then all entries except those in the IP exclude list will be added to the database. Otherwise only Ips that match the list will be inserted.

An Example of this file is:

```
# Include file IP mask list
# Ips can be in the format of a.b.*.*
10.0.*.*
192.127.*.*
```

## IpExcludeListFile.txt

The conf/IpExcludeListFile.txt file contains a list of IP masks that will not be added to the discovery database. If no entry given then all entries that match the IP Include list will be added to the database.

```
# Exclude file IP mask list
# Ips can be in the format of a.b.*.*
130.0.*.*
140.127.*.*
```

# The Discovery & Provision process logic

---

The Inventory, Network Discovery & Provisioning product is a very flexible discovery & provisioning product, with very strong tailoring capabilities.

Please refer to the “**Discovery Users Guide**” for a detailed explanation on how to extend the discovery capabilities of the product, and how to extend its scheme, to the one you exactly need.

## The discovery process

The Discovery process starts with a provisioning file that is supplied by the user.

The information in the provisioning file instructs the '**discovernet**' command what nodes to discover, using which protocol.

The product by default only discovers the nodes which are in the provisioning file.

However, in the discovery process the ARP table of a given node is learned, and there is an option that moves the newly discovered ARP entries to the provisioning file. Using the combination of learning the nodes in the provisioning file, and adding the ARP entries into the provisioning file, we are performing a Layer3 discovery process.

In this case you will put in the provisioning file, only anchor nodes (such as the main router in every site of your organization, and the discovery of the entire network will be performed automatically, as the cycles of discovery are continued.

### ***Limiting the discovery range***

The product has 2 configuration files which can limit the discovery process, and prevent learning addresses outside the scope of the desired network.

The files are:

- IpIncludeListFile.txt – which specifies the ranges of IPs that should be learned.
- IpExcludeListFile.txt – which specifies the ranges of Ips that should not be learned.

## *The current discovery table scheme*

following is the current tables scheme of the discovery related tables.

Some of Tables are acting as sparse matrix, when not all the fields are filled for all objects.

Name prefix conventions:

str – refers to string

n – refers to an Integer

## **tblObject**

TblObject is the main table that holds the discovery information.

It contains entries from several type:

- Node – A network node entry
- Interface – An Interface entry (each SNMP discovered node) has one or more interfaces.
- Address – An Address entry, maps an IP address to a Node. A node might have several IP addresses.

Field Name	Usage
nObjectId	A unique ID of the object
nParentObjectId	The ID of the parent of the current object, when relevant
strMacAddr	The Mac address of an Interface
strType	The type of the object: Node, Interface, Address
strSysObjectId	The SysobjectId of the node
strSysLocation	SysLocation of the Node
strSysContant	SysContact value of the node
strDiscoveryMethod	The discovery method used: ICMP   SNMP
strMonitoringMethod	The monitoring method used:ICMP   SNMP (used with the Monitoring Platform product).
nSNMPVersion	SNMP version discovery method for the Node (1 2 3)
strCommunity	The community value used for accessing the device (in SNMP V3 – the user name)
strPassword	The password for SNMP V3 access
nTimeoutMsec	Timeout value for requests
nRetries	Retry count when timeout occurs
nIfIndex	If Index in Interface definitions
strDescription	SysDescription for nodes, and IfDescriptions for Interfaces
strSysName	SysName of Node
nUpdateTime	Last time the record was updated
strIfAlias	The IF Alias of an Interface
nIfType	ifType value for an Interface
strIfName	IfName for for an Interface
strDNSName	for an Address entry
strSubnetMask	The subnet mask of the given 'Address' entry
strSubnetAddr	the subnet asddress for a given 'Address' entry
strServiceLevel	A documentation field, that defines the service level of this node (used with the monitoring platform product)
strGroupingLevel1	A documentation field, that defines the highest level of grouping of the node (used with the monitoring platform product)
strGroupingLevel2	A documentation field, that defines the 2nd level of



	grouping of the node (used with the monitoring platform product)
strGroupingLevel3	A documentation field, that defines the 3rd level of grouping of the node (used with the monitoring platform product)
strGroupingLevel4	A documentation field, that defines the 4 <sup>th</sup> level of grouping of the node (used with the monitoring platform product)

## tblArp

tblArp contains the ARP table of the SNMP discovered nodes.

Field Name	Usage
strIpAddr	The IP address of the entry
strMacAddr	The MAC address related to the given IP
nUpdateTime	The last discovery time
nLastDiscoveredObjectId	The node on which the entry was discovered

## tblVlanInfo

tblVlanInfo contains VLAN information.

Field Name	Usage
nIfIndex	The IfIndex of interface
strVlanId	The Vlan ID for the given interface.
nUpdateTime	The last discovery time
nNodeObjectId	The node on which the entry was discovered

## tblObjectLastMonitoringTime

tblObjectLastMonitoringTime contains the time of the last run of the discovery process.

Field Name	Usage
nLastDiscoveryTime	The last discovery time

## Discovery Methods

There are currently 2 Discovery methods:

- SNMP – An SNMP based discovery method, used to load snmp information into the database.
- ICMP – An ICMP discovery method. Used for inserting data in to the the databse. It only fills the DNS of the node, if there is one, and fills the default Interface and Address values. This discovery method is usually used with the Monitoring Platform product.

Note that you can alter the discovery rules, however the product only supports these methods

## The Discovery command lines

---

The Discovery process defined by Inventory, Network Discovery & Provisioning 3.0 is based on a set of commands, which get provisioning information from the user.

❗ All the commands described in this section have a lock mechanism that prevents them to have multiple instances of the program running concurrently.

### *discovernet*

This command line is the main function that performs the discovery actions as described in the previous chapter.

The following is the output of the command when no parameters are supplied.

Usage:discovernet Action [ Provisioning\_file ]

Where: Action = [ Add | DeleteNotInProvisioning | DeleteProvisioningList  
| AddProvisioningFromArp | Periodic | ForceRediscover | All ]

Add - adds nodes which are not in the provisioning file

DeleteNotInProvisioning - deletes nodes which are in the database and not in the provisioning file

DeleteProvisioningList - deletes nodes which are in the provisioning list

AddProvisioningFromArp - add entries from arp table to existing provisioning file

Periodic - rediscover all nodes which their last discovery time expired

ForceRediscover - rediscovers all nodes which are flagged for rediscovery in the provisioning file

All - does all the actions listed above not including:

DeleteProvisioningList & AddProvisioningFromArp

With this command all discovery operations can be perform.

## *buildprovisioningfromarp*

This command line performs only the operation of taking the arp table and filling the conf/provisioning.csv with the enties found in it and not in the provisioning file.

The Template of the line added for every IP found in the tag <ProvisioningParametersDefault> in the conf/config.xml

The following is the output of the command when no parameters are supplied.

```
Usage:buildprovisioningfromarp Provisioning_file
```

```
add entries from arp table to existing provisioning file
```

## *discovernetworkwitharp*

This command line performs 2 operations:

1. It runs the buildprovisioningfromarp, so in the next discovery it will discover new nodes found in the last discovery process.
2. It runs the discovernet with the parameters supplied by the user.

This command can be the base of a cyclic network discovery process.

The following is the output of the command when no parameters are supplied.

```
Usage:discovernetwitharp Action [ Provisioning_files_dir ]  
      Where: Action = [ Add | DeleteNotInProvisioning | DeleteProvisioningList  
                      | AddProvisioningFromArp | Periodic | ForceRediscover | All ]  
      Add - adds nodes which are not in the provisioning file  
      DeleteNotInProvisioning - deletes nodes which are in the database and  
                              not in the provisioning file  
      DeleteProvisioningList - deletes nodes which are in the provisioning list  
      AddProvisioningFromArp - add entries from arp table to existing  
                              provisioning file  
      Periodic - rediscover all nodes which their last discovery time expired  
      ForceRediscover - rediscovers all nodes which are flagged for rediscovery  
                      in the provisioning file  
      All - does all the actions listed above not including:  
            DeleteProvisioningList & AddProvisioningFromArp
```

## *inventory*

This command line is a generic “**discovery based inventory & provisioning command**”. It runs discovery rules that are defined using the discovery rules definition gui.

```
Usage:inventory [root_discovery_item_name [parameters_file]]  
Where:  
root_discovery_item_name – the name of the discovery rule that acts as the root  
of the discovery & provisioning instructions operations  
  
parameters_file – a file name containing a list of key=values lines that will be  
used as external parameters to the discovery rules run by the command.  
  
Please look at the discovery user manual on how to pass parameters.
```

The command can run without parameters. In this case it runs on all the discovery rules found in [INSTALLDIR]/conf/discoveryrules and makes sure that the database fits the tables and fields defined in the discovery rules listed there.

## Utility commands

---

### *snmptrap*

The snmptrap command sends a trap to a target SNMP manager based on the parameters passed in the command line

Usage: snmptrap strIpAddr, nPort, strCommunity, strEnterprise, nGeneric, nSpecific, strOID, stringValue

Example: snmptrap 127.0.0.1 162 public  
.1.3.6.1.4.1.9.17888.1.2.1 4 5 1.3.6.1 'trap message'

### Parameter Description

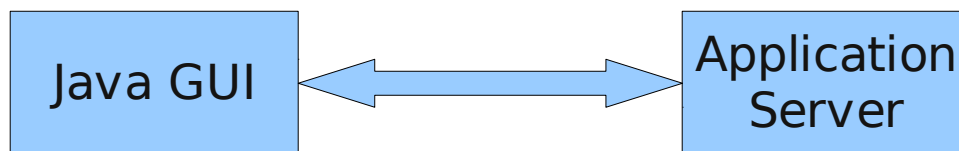
- strTargetIp - IP address of traps receiver.
- nTargetPort - Port number of traps receiver.
- strCommunity - Community of traps receiver.
- strEnterprise - Enterprise OID of sent trap.
- nGeneric - Generic number of trap.
- nSpecific - Specific number of trap.
- strOID - oid of variable
- strValue - string value of variable

# The Inventory, Network Discovery & Provisioning GUI

---

The Inventory, Network Discovery & Provisioning product comes with a Java based Graphical User Interface (GUI). The GUI can run on any platform that supports Java.

The Client uses HTTP(s) to access the application server. It can be located remotely to the product's core.



## *The product menu*

The products menu has the following structure

- Administration – The main administration sub menu
  - Edit Product Config File – opens the product's conf/config.porp file, and allows to edit it
  - Edit Provisioning File – opens the default provisioning file, pointed by the conf/config.xml file
  - Edit GUI Config File – opens the GUI's conf/configGUI.porp file, and allows to edit it
  - License info – shows the current license information -
  - Edit Connection File – shows the local connections information. Please refer to the security chapter for more information
  - Permissions Management – The permissions management sub menu
    - Users Management – opens a query panel with the users information. Please refer to the security chapter for more information
    - Commands permission management – opens a query panels with the

commands permissions information. Please refer to the security chapter for more information

- Exit – exit thg GUI
- Run – The sub-menu that handles the discovery operations.
  - Run Discovery – This menu item launches the discovernet command with the ALL option.
  - Add Arp Info to Provisioning File – This menu item launches only the “Add Arp Info to Provisioning File” option of the “discovernet” command line.
  - Run Arp Provisioning and Discovery – This menu items run a command line that performs the 2 actions of adding the arp table to the provisioning file, and performing the discovery.
  - Discovery Jobs Status – This menu item, launches a panel, with the jobs status of the jobs launched by this sub-menu.
- Reports – This sub-menu lists all the reports of the product.
  - Nodes – This report lists all the discovered nodes
  - Interfaces – This report lists all the Interfaces discovered on the SNMP discovered nodes.
  - Vlans – This report lists the vlans discovered
  - Arp – This report lists the ARP entries discovered
  - All Jobs Status – This report lists all the jobs in the scheduler's db. For more information look in the scheduler chapter.
- Help – A sub menu with help information
  - About – Shows the product information
  - Documentation – Displays this documentation



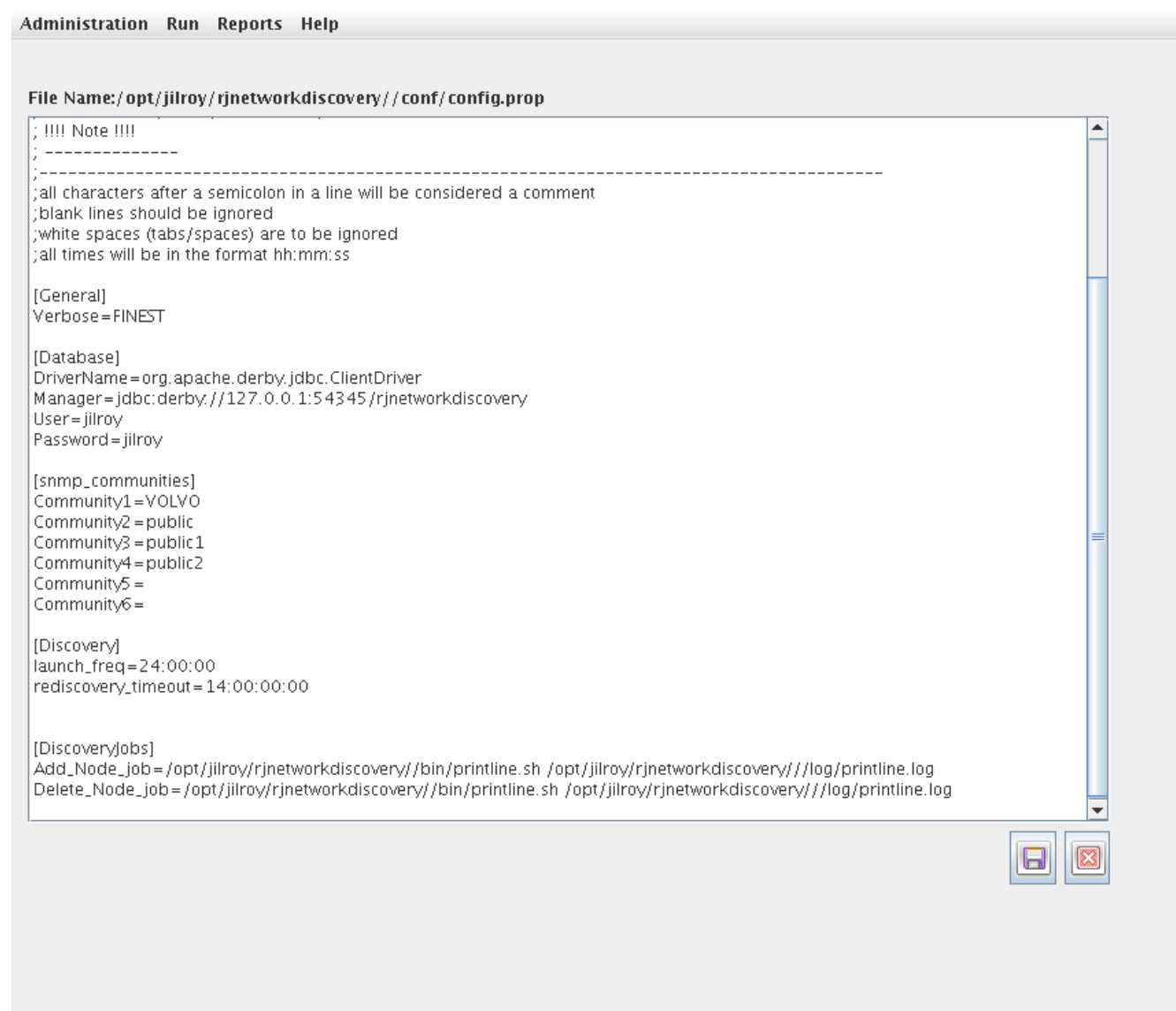
## *The Product's Panels*

### **Edit Product Config File**

This Panel will only function for the administrators role.

It displays the conf/config.prop file in the product's install directory.

It is reached from: Administration -> Edit Product Config File



The supported options are:

- Save – to save the changes done in the file
- Cancel – to leave the screen.

## Edit Provisioning File

This Panel will only function for the administrators role.



It displays the default provisioning file pointed by conf/config.xml file. Its name is: conf/provisioning.csv file in the product's install directory.

It is reached from: Administration -> Edit Provisioning File

**Administration** Run Reports Help

**File Name:** /opt/jilroy/rjnetworkdiscovery//conf/provisioning.csv

```
#format: single line provisioning entries
#-----
#ip_addr,discovery_method,MonitoringMethod,ForceDiscovery,community,password,Snmpversion,TimeoutMsec,Retry,ServiceLevel,GroupingLevel1,GroupingLevel2,GroupingLevel3,GroupingLevel4
#
# where:
# discovery method=none|SNMP
# monitoring method=ICMP|SNMP
# force discovery=0|1
# community acts as a userid for snmpv3
# password is only relevant for snmp v3
# snmp version=1|2|3
# ServiceLevel=(optional)Any Category of service level (Platinum, Gold, Silver,Bronze,...)
# GroupingLevel1 - GroupingLevel4= (optional) Grouping level
#                               for example location      : 1- state, 2-city, 3-street, 4-building
#                               or                          : company, state, city, department
10.0.1.201,ICMP,ICMP,0,none,none,1,3000,3,Gold,IBM,France,Paris,Marketing
10.0.1.202,ICMP,ICMP,0,none,none,1,3000,3,Gold,IBM,England,London,Finance
10.0.1.203,ICMP,ICMP,0,none,none,1,3000,3,Platinum,Jilroy,US,NY,Management
10.0.1.254,SNMP,SNMP,1,VOLVO,none,1,3000,3,Platinum,Jilroy,US,NY,RND
127.0.0.1,ICMP,ICMP,0,none,none,1,3000,3,Gold,IBM,France,Paris,Marketing
```



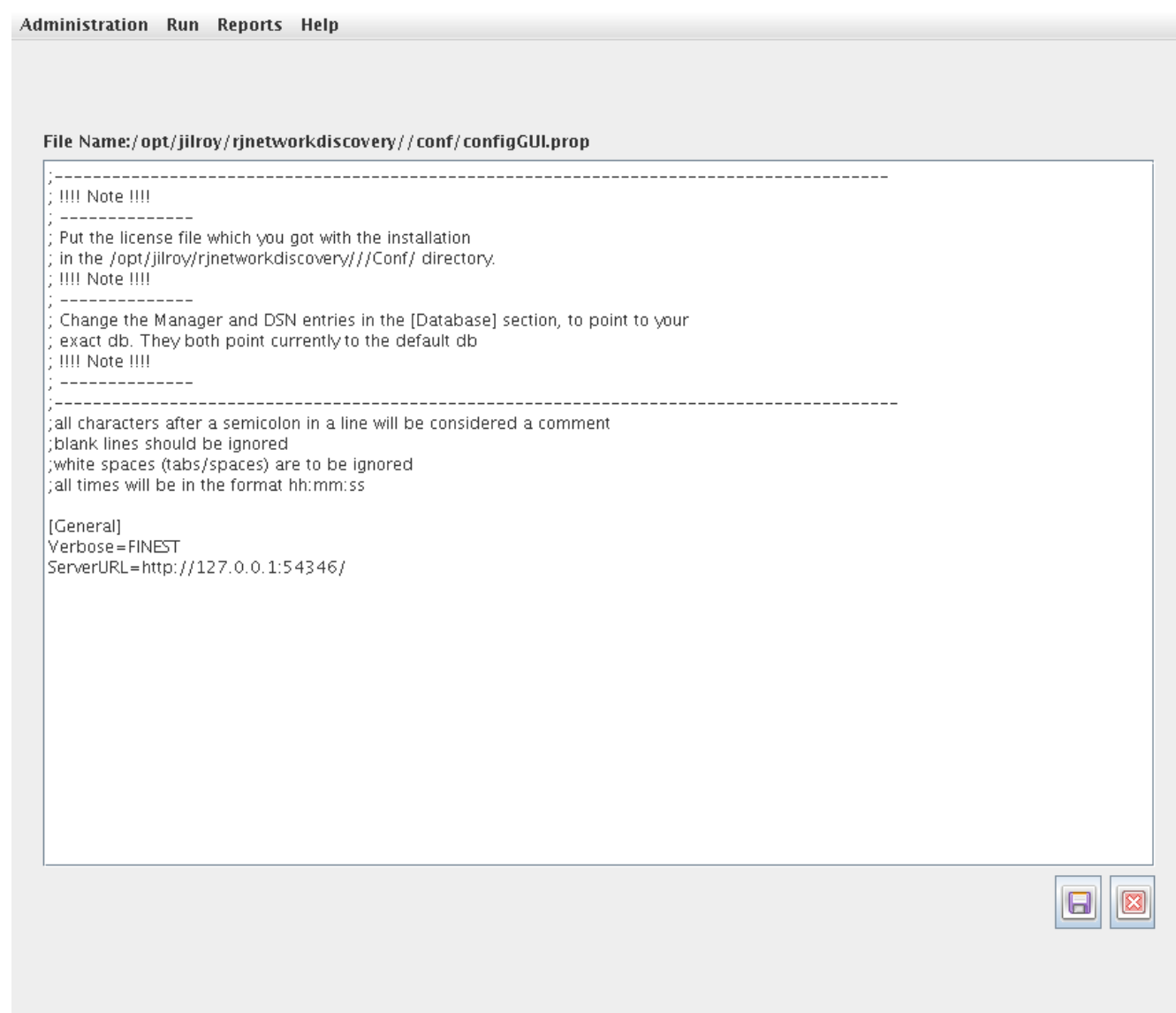
The supported options are:

- Save – to save the changes done in the file
- Cancel – to leave the screen.

## Edit GUI Config File

This Panel displays the conf/configGUI.prop file in the GUI's install directory.

It is reached from: Administration -> Edit GUI Config File



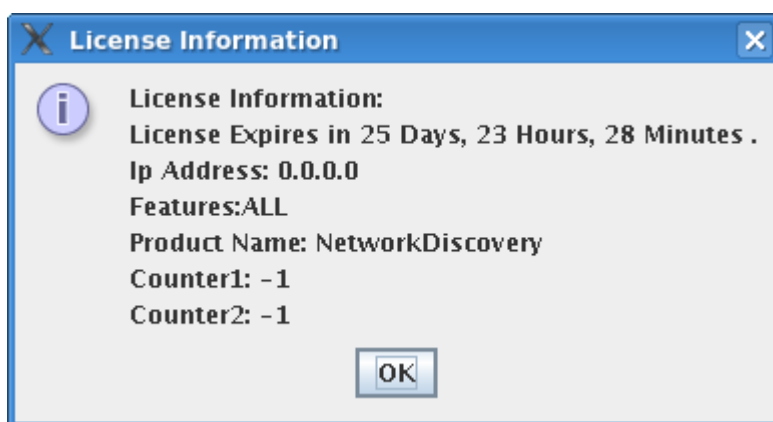
The supported options are:

- Save – to save the changes done in the file
- Cancel – to leave the screen.

## License Info

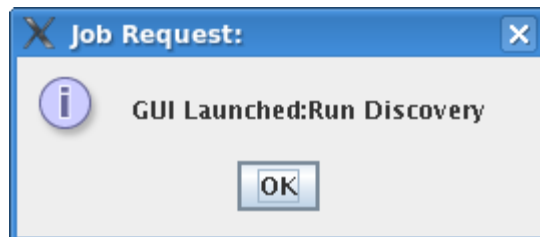
This option will a message box with the current license information.

It is reached from: Administration -> License Info



## Run Discovery commands

The options of run a discovery job will output this kind of message box.









You will be able to see the status of the jobs in the "Discovery Jobs Status" panel.

## Discovery Jobs Status

This Panel shows the status of the commands launched by the run sub-menu.  
It is reached from: Run -> Discovery Jobs Status

Administration Run Reports Help  
Report Name: Discovery Jobs Status

Query Fields:  
Job Description  Job Id


Actions:  
 Kill/Delete Type:       Lines Limit:

Query Results:

JOB_ID	JOB_NAME	JOB_CATEGORY	DESCRIPTION	JOB_COMMAND	START_TIME	STOP_TIME	Select
61	Run Discovery	Default	GUI Launched Run Disco...	/opt/jilroy/rjnetworkdisc...	Fri Apr 25 15:54:56 IDT...	null	<input type="checkbox"/>

The Panel shows the selected jobs based on the following query fields:

- Job description
- Job Id

 The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblJobs database table.

Actions:

- refresh – performs the query again
- kill selected job

- Delete selected job – marks the job for deletion. It will be deleted after it ends, by the scheduler process itself
- Restart selected job – restart the selected job, only if it already ended.
- Select All – select all jobs returned.
- Un-select ALL – unselect all selected jobs.
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## Nodes Report

The nodes report lists all the nodes discovered by the Network Discovery module.

Administration Run Reports Help




Report Name: Nodes

Query Fields:

IP Address:  DNS Name:  Node SysObjectId:

Discovery Method:  Monitoring Method:

Actions:


   Lines Limit:

Query Results:

NODE_OBJECTID	NODE_IP	DNS_NAME	SYSDESCRIPTION	SYSOBJECTID	Select
2	10.0.1.203				<input type="checkbox"/>
4	127.0.0.1				<input type="checkbox"/>
6	10.0.1.201				<input type="checkbox"/>
8	10.0.1.254		Cisco IOS Software, C35...	1.3.6.1.4.1.9.1.615	<input type="checkbox"/>
59	10.0.1.202				<input type="checkbox"/>

The Panel shows the selected Nodes based on the following query fields:

- IP Address
- Discoverey Method
- DNS Name
- Monitoring Method
- Node's SysObjectId

 The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblObject database table.

## Actions:

- refresh – performs the query again
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## Interfaces Report

The Interfaces report lists all the interfaces discovered by the Network Discovery module.



**Administration** **Run** **Reports** **Help**

**Report Name:** Interfaces

**Query Fields:**

**IP Address:**  **Interface Alias**  **Interface Description**

**Actions:**

  **Lines Limit:**

**Query Results:**

IF_OBJECTID	NODE_IP	IF_INDEX	IF_DESCRIPTOR	IF_NAME	IF_ALIAS
9	10.0.1.254	3	Vlan3		
10	10.0.1.254	10108	GigabitEthernet0...		
11	10.0.1.254	10122	GigabitEthernet0...		
12	10.0.1.254	10111	GigabitEthernet0...		
13	10.0.1.254	5011	Port-channel11		
14	10.0.1.254	10105	GigabitEthernet0...		
15	10.0.1.254	10127	GigabitEthernet0...		
16	10.0.1.254	10119	GigabitEthernet0...		
17	10.0.1.254	10112	GigabitEthernet0...		
18	10.0.1.254	6	Vlan6		
19	10.0.1.254	1	Vlan1		
20	10.0.1.254	10110	GigabitEthernet0...		
21	10.0.1.254	10121	GigabitEthernet0...		
22	10.0.1.254	10118	GigabitEthernet0...		
23	10.0.1.254	5009	Port-channel9		
24	10.0.1.254	5	Vlan5		
25	10.0.1.254	10128	GigabitEthernet0...		
26	10.0.1.254	10113	GigabitEthernet0...		
27	10.0.1.254	10107	GigabitEthernet0...		
28	10.0.1.254	4	Vlan4		
29	10.0.1.254	10501	Null0		
30	10.0.1.254	10106	GigabitEthernet0...		



The Panel shows the selected Interfaces based on the following query fields:

- IP Address of the node
- Interface IFAlias
- Interface Description

❗ The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblObject database table.

Actions:

- refresh – performs the query again
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## Vlans Report

The Vlans report lists all the vlans discovered by the Network Discovery module, and the interfaces that belong to them.



Administration
Run
Reports
Help

Report Name: Vlans

Query Fields:

IP Address:  Interface Alias  Interface Description

Actions:



Lines Limit:

Query Results:

NODE_IP	VLAD_ID	IF_INDEX	INTERFACE_D...	IF_NAME	IF_ALIAS
10.0.1.254	0	5008	Port-channel8		
10.0.1.254	1	10125	GigabitEthernet0...		
10.0.1.254	1	10126	GigabitEthernet0...		
10.0.1.254	1	10127	GigabitEthernet0...		
10.0.1.254	1	10128	GigabitEthernet0...		
10.0.1.254	2	10105	GigabitEthernet0/5		
10.0.1.254	2	10106	GigabitEthernet0/6		
10.0.1.254	2	10107	GigabitEthernet0/7		
10.0.1.254	2	10113	GigabitEthernet0...		
10.0.1.254	2	10114	GigabitEthernet0...		
10.0.1.254	21	10101	GigabitEthernet0/1		
10.0.1.254	6	10108	GigabitEthernet0/8		
10.0.1.254	6	10109	GigabitEthernet0/9		
10.0.1.254	6	10110	GigabitEthernet0...		
10.0.1.254	6	10111	GigabitEthernet0...		
10.0.1.254	6	10112	GigabitEthernet0...		
10.0.1.254	8	10102	GigabitEthernet0/2		
10.0.1.254	8	10103	GigabitEthernet0/3		
10.0.1.254	8	10104	GigabitEthernet0/4		

The Panel shows the selected Vlans based on the following query fields:

- IP Address of the node
- Interface IFAlias
- Interface Description

❗ The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblObject database table.

Actions:

- refresh – performs the query again
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## ARP Table Report

The ARP report lists all the ARP entries discovered by the Network Discovery module, and the nodes they were last discovered on.



Administration Run Reports Help

Report Name: Arp

Query Fields:

Arp IP Address:  Arp Mac Address:  Last Discovered Node Address:

Actions:

  Lines Limit:

Query Results:

DISCOVERED_ADDRESS	STRMACADDR	LAST_DISCOVERED_ON
10.0.32.92	0X0015609FA658	10.0.1.254
10.0.16.168	0X000D9DD08309	10.0.1.254
10.0.1.182	0X000D9DD080D9	10.0.1.254
10.0.0.25	0X000E7F304321	10.0.1.254
10.0.0.24	0X000FFE0B2560	10.0.1.254
10.0.0.199	0X00E08602D842	10.0.1.254
10.0.16.70	0X000E7F65B217	10.0.1.254
10.0.32.71	0X000E7F653501	10.0.1.254
10.0.1.89	0X000FFE0D807D	10.0.1.254
10.0.0.15	0X0002B389E396	10.0.1.254
10.0.1.67	0X00110A989D2D	10.0.1.254
10.0.16.254	0X0013601723C4	10.0.1.254
10.0.0.53	0X00D068078A72	10.0.1.254
10.0.16.89	0X000B6AF41FD9	10.0.1.254
10.0.1.235	0X000C290C751F	10.0.1.254
10.0.1.94	0X000C295CB588	10.0.1.254
10.0.0.250	0X0012DAC6B5C0	10.0.1.254
10.0.0.35	0X000802DF7C9B	10.0.1.254
10.0.2.67	0X000129742100	10.0.1.254
10.0.1.185	0X0001029DF2F6	10.0.1.254
10.0.2.10	0X000E7F2E3141	10.0.1.254
10.0.0.225	0X001B781C9DCD	10.0.1.254

The Panel shows the selected ARP entries based on the following query fields:

- IP Address of the node they were last discovered on
- ARP IP address
- ARP Mac Address

❗ The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

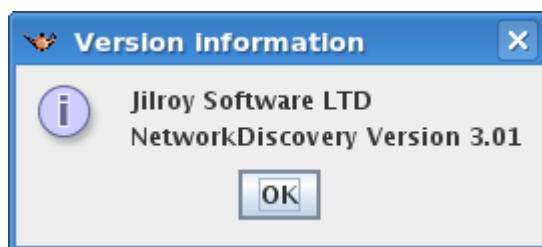
The fields are the ones described in the tblObject database table.

Actions:

- refresh – performs the query again
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of raws returned.

## About

The About menu option lists the product version information.



## Backup and Restore for the Database

---

The product supports a way to backup its database into CSV files, and restoring the tables from the CSV files.

### *exporttable2csv*

This command exports a table to a CSV file, that can be later loaded back to the database using a supplied utility

The following is the output of the command when no parameters are supplied.

```
format: exporttable2csv csv_file_name
```

where the *csv\_file\_name* is the full path to the file with instructions about the operation requested.

The file structure is:

- Line 1:Table name
- Line 2:CSV File full path
- Line 3:Separator String
- Line 4:Replace separator string

If separator is found in a table field then it replaces it with the replace string

### *importcsv2table*

This command imports a CSV file in to a table.

The following is the output of the command when no parameters are supplied.

`format: importcsv2table csv_file_name`

where the *csv\_file\_name* is the full path to the file with instructions about the operation requested.

The file structure is:

- Line 1:Table name
- Line 2:CSV File full path
- Line 3:Separator String
- Line 4:Replace separator string

If replace separator is found in a table field then it replaces it with the string. The same CSV file can be used for the export and import file.

## Additional SQL related commands

---

The product is highly based on the database. It has several user commands which are related to the database.

### *Query analyzer for the database*

In order to drill down into the database, and perform SQL queries on the product's database, we recommend to use DbVisualizer. Its free version is good enough for all required actions.

### *createdb*

This command is creating the product's database. It extracts the relevant information about the name of the database from the conf/config.xml file and from the env.db script.

```
format: createdb  config.xml
```

### *createtables*

This command creates the product's tables. It extracts the tables information from the definition files located in the /conf/tables/ directory.

```
format: createtables  config.xml
```

### *dbconf*

This command is recreating the database. It is used without parameters. It can be reused when there are problems with the database.



## *loadtableinfo*

This command is used to load data into a given table. It is similar to the `importcsv2file`, but is easier to use

`format: loadtableinfo table_name csv_file_name separator`

Where

1. `Table_name` - The table name which will be update
2. `csv_file_name` – The full path to the CSV file containing the data
3. `separator` – The separator string, in the csv file

## *sqlexec*

This command is used to execute SQL commands on the database form a batch file.

`format: sqlexec sql_commadns_file_name`

The sql commands file requires that every command will be in a single line.

## *startdb*

This command starts the database. It should be added to the crontab on the computer startup. It is currently tailored to use the hsql DB.

## *stopdb*

This command stops the database. It is currently tailored to user the hsql DB.

## *truncatetables*

This command line truncates all the tables defined in the conf/tables configuration files.

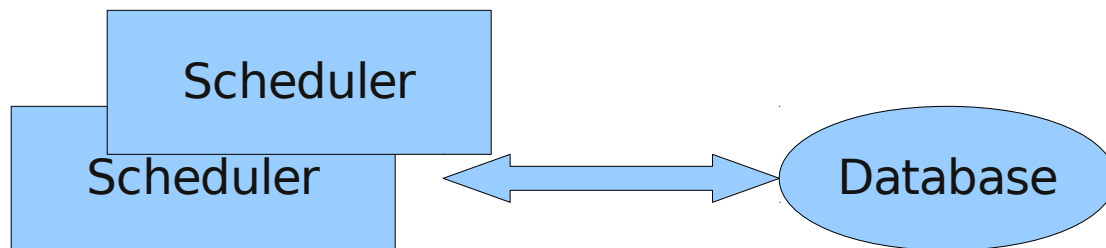
Usage:truncatetables ALL table_name
-------------------------------------

# The product Scheduler

---

The product comes with a Scheduler process, which handles the launching of the product related services and its event handlers.

This scheduler is based on the database and manages its scheduling based on the information stored in the database.



The scheduler runs jobs. A job is a single autonomic unit of work. By default the scheduler runs the jobs as external processes asynchronously, however there are commands that are run internally by the scheduler as part of the scheduler code.

The scheduler comes with tools to schedule jobs and to manage the scheduled jobs:

- Scheduler GUI
- Scheduler commands

The scheduler was designed in a way that will enable to limit the number of jobs that will run concurrently.

The scheduling process was designed for scale. It is possible to split commands between multiple schedulers running on different machines in-order to improve performance.

## *The Scheduler's tables structure*

### **tblJobs**

tblJobs holds the scheduled jobs information.

Field Name	Usage
nSchedulerId	The Id of the scheduler that should process the current job.
nJobId	The ID of the Job
strCommand	The command line that is executed
strStatus	This field can contain the current status of the job, if the job it self supports its update
strName	The name of the job
strDescription	A description of the jobs work
nScheduleTime	The time the job should start (if null, then it will start immediately)
nStartTime	The time the job started (or NULL)
nStopTime	The time the job ended (or NULL)
nRestartFrequencyMin	Determines if the job should be restarted and if yes, how long after it has finished. The time is in minutes
strJobCategory	The category that the job belongs to
fDelete	Indicates if the job should be deleted from the table by the scheduler (values:0,1)

### **configDbScheduler.xml**

The conf/configDbScheduler.xml file is an internal configuration file, that would hardly have to be processed by the user. However this is the main configuration file passed as a parameter to the scheduler process.

In order to change the database that the scheduler uses, change the definitions in

the conf/config.prop file.

## Categories file

The Scheduler has a mechanism for preventing over-flooding of the system with event-related jobs. The mechanism is based on Job category classes. You can define the maximal amount of concurrent jobs that should run from a given category class. If the full capacity of the class has been reached, then the additional jobs are queued until the previous ones end.

The classes are defined in the file conf/categories.txt:

## Categories.txt file

```
# Scheduler categories
Default,10
DiscoveryEvents,10
```

A remark line starts with #

The format of every raw is:

- Category name
- Maximum number of concurrent jobs from the given category.

After changing the file the Scheduler must be restarted.

## GUI








Administration Run Reports Help

Report Name: All Jobs Status

Query Fields:

Job Category:  Job Description:  Job Id:  Job Command:

Actions:

 Kill|Delete Type:        Lines Limit:

Query Results:


JOB_ID	JOB_NAME	JOB_CAT...	DESCRIP...	JOB_CO...	START_T...	STOP_TL...	Select
1	discovernet...	Default	discover net...	/opt/jilroy/r...	Fri Apr 25 1...	Fri Apr 25 1...	<input type="checkbox"/>
61	Run Discovery	Default	GUI Launche...	/opt/jilroy/r...	Fri Apr 25 1...	null	<input type="checkbox"/>

This panel is reached through:

Reports->All Jobs Status

The Panel shows the selected jobs based on the following query fields:

- Job category
- Job description
- Job Id
- Job Command

 The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblJobs database table.

#### Actions:

- refresh – performs the query again
- kill selected job
- Delete selected job – marks the job for deletion. It will be deleted after it ends, by the scheduler process itself
- Restart selected job – restart the selected job, only if it already ended.
- Select All – select all jobs returned.
- Un-select ALL – unselect all selected jobs.
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## Internal commands

There is currently only one internal command recognized by the scheduler.

### **snmptrap**

The snmptrap command sends a trap to a target based on the parameters passed in the command line. Look at the utilities chapter for a description on the parameters of this command.

## Loading jobs to the database

The Scheduler comes with a command that enables it to load jobs into the tblJobs.

### loadjobsfromfile

This command load an XML file containing jobs information into the tblJobs.

```
Usage:loadjobsfromfile Jobs.xml.file
```

The XML file format is:

```
<ProcessList>
  <JobEntry>
    <strJobName>discovernetwitharp</strJobName>
    <strJobDescription>job description</strJobDescription>
    <strJobCategoryName>Default</strJobCategoryName>
    <strDelayTime>00:00:00</strDelayTime>
    <strJobCommandLine>command full path</strJobCommandLine>
    <strRestartFrequencyMin>24:00:00</strRestartFrequencyMin>
    <nSchedulerId>1</nSchedulerId>
  </JobEntry>
</ProcessList>
```

### ProcessList

This is a wrapper tag for all entries of in the XML file

### JobEntry

This tag contains the jobs definition tags



**strJobName**

The name of the job

**strJobDescription**

The job description

**strJobCategory**

The name of the job category

**strDelayTime**

The number of minutes after the load, that the job will be able to start.

**strJobCommandLine**

the full command line of the job

**strRestartFrequencyMin**

The time in minutes after the job end, that it should be restarted again

**nSchedulerId**

The number of the scheduler that will run the job.

❗ If the job has to update its status in the tblJobs strStatus field, you can pass to the job aparameter with the value @JobId@, that will be replaced with the Job Id. You can use this information to know the exact Job entry.

## The product Security mechanism

---

The product has a security mechanism implemented in its Application Server (tomcat) and implemented through the GUI.

The permissions mechanism is built on the following concepts:

- Users belong to given roles
- Certain commands can be executed only by certain roles.

The authentication information is passed with each request by the GUI, based on a connection\_info file held in the conf/connection.prop file.

### *The Users and Permissions table*

In order to implement the security mechanisms 2 tables were added to the products database.

#### **tblUsers**

tblUsers contains the Users and Roles information

Field Name	Usage
strUser	The User Name
strPassword	The User's Password
strRole	The Role of the user.

A user can belong only to a single role.

### Default values

The product comes with the admin user defined, where its password is admin, and its role is administrator.

Do not delete this entry, but you are allowed to change the password of the administrator.

## **tblPermissions**

tblPermissions contains the information about the permitted commands for a given role

Field Name	Usage
strCommand	The Command Name
strRole	A Role that has of the user.

A given command can be allowed for multiple roles.

### Default values

The product comes with the @ALL@ entry with the role administrator, which defines that the administrator can execute all commands.

Do not delete this entry.

## *The connection.prop file*

This file contains the security information of the current user.

The format of the file is:

```
[General]
User=admin
Password=admin
EncryptionKey=0123456789abcdef
```

## **User**

The User name of the GUI user.

## **Password**

The password of the GUI user.

## **EncryptionKey**

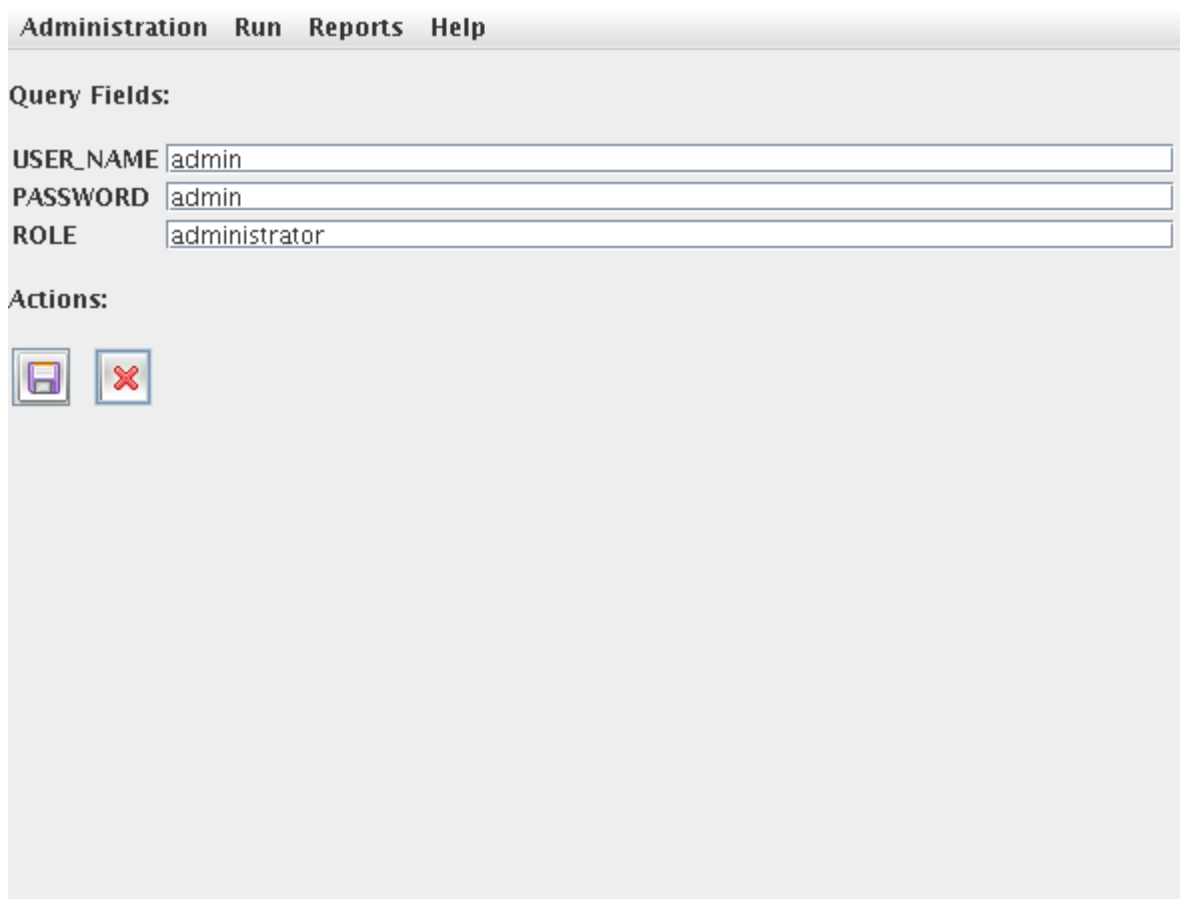
An optional parameters when there is communication between peers when the application server is acting as a relay.

This parameter is not relevant with most of our products.

## *The Security GUI*

The product comes with several screens to manage the security mechanism  
Client Side

### **connection information management**



The screenshot shows a web application interface with a menu bar at the top containing 'Administration', 'Run', 'Reports', and 'Help'. Below the menu bar is a section titled 'Query Fields:' containing three input fields: 'USER\_NAME' with the value 'admin', 'PASSWORD' with the value 'admin', and 'ROLE' with the value 'administrator'. Below these fields is a section titled 'Actions:' containing two icons: a save icon (a floppy disk) and a cancel icon (a red X).

This panel is reached through:  
Administration->Edit Connection File

The fields are the ones described in the conf/connection.prop

Actions:

- Save – save values in the conf/connection.prop file
- Cancel – End without doing anything

## Server Side

### the users definition panel

This screen is used to define the users of the system.

The screenshot shows a web-based interface for user management. At the top, there is a navigation bar with links: Administration, Run, Reports, and Help. Below this, the title 'Report Name: Users management' is displayed. The 'Query Fields' section contains two input fields: 'User:' and 'Role:'. The 'Actions' section includes five icons: a refresh icon, a plus icon, a minus icon, a home icon, and a printer icon. To the right of these icons is a 'Lines Limit' input field set to '1000'. Below the actions is the 'Query Results' section, which contains a table with the following data:

USER_NAME	PASSWORD	ROLE	Select
admin	admin	administrator	<input type="checkbox"/>

This panel is reached through:

Administration->Permissions Management -> Users Management

The Panel shows the selected users based on the following query fields:

- User – User name
- Role – Role Name

**i** The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .

The fields are the ones described in the tblUser database table.

Actions:

- refresh – performs the query again
- Add - adds a new entry to the users table
- Delete – Delete the selected entries from the users table
- update – Update the first selected entry from the users table
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of rows returned.

## the Commands permissions panel

This screen is used to define the commands permissions. It defines the roles that can execute given commands.

This panel is reached through:

Administration->Permissions Management ->  
Commands Permissions Management

The Panel shows the selected users based on the following query fields:

- Command – Command name
- Role – Role Name

**i** The values entered in the query fields can be prefixes. To enter a prefix or a wild character write the string you want with % at its end or at the location of the wild character .





Administration Run Reports Help

Report Name: Commands Permission management

Query Fields:

Command:  Role:

Actions:

    Lines Limit:

Query Results:

COMMAND	ROLE	Select
@ALL@	administrator	<input type="checkbox"/>



The fields are the ones described in the tblUser database table.

Actions:

- refresh – performs the query again
- Add - adds a new entry to the users table
- Delete – Delete the selected entries form the users table
- Print to CSV – put query result into a CSV file for later processing
- Lines limit – sets the max number of raws returned.

## *General Commands permitted*

Following is a list of all the common application server commands:

### **RRD**

A code for allowed the RRD related commands.

### **Reports commands**

For each Report the user ask, you can define a security entry(It applies also to your extensions – see the GUI programmers guide). The name you specify as the query name in the form definition.

# Final Page

---

## ***More Information***

More information about Jilroy Software, the Inventory, Network Discovery & Provisioning product, and our other products can be found on our web site.

[www.jilroy.com](http://www.jilroy.com)

## **Contact Us**

For any information or problem, request for information or extension idea related to the Inventory, Network Discovery & Provisioning or any other product, please contact one of the following

email addresses.

## **Sales**

[sales@jilroy.com](mailto:sales@jilroy.com)

## **Product Management**

discovery\_[pm@jilroy.com](mailto:pm@jilroy.com)

## **FTP Site**

All jilroy products can be downloaded from our Web site. Please contact [info@jilroy.com](mailto:info@jilroy.com) for download information.