



NeuBiro
User's guide

2017-09-14

Version 0.7

Table of contents

1. Introduction	1
1.1. Technological stack	1
2. Getting started	2
2.1. Main dependencies	2
2.2. How to install	2
2.2.1. Installation steps	3
2.2.2. Installation folder's layout	8
3. Working with NeuBiro	10
3.1. Overview of the working pipeline	10
3.2. The user interface	10
3.3. Configuration	10
3.4. Importing the data set	11
3.4.1. How to prepare the data set for NeuBiro	13
3.5. Running the analysis	13
3.5.1. Advanced analysis	14
3.5.2. Organization of the work folder	14
3.6. Transfer the aggregated data	14
3.7. The logging tab	15
4. Conclusion	16
5. FAQ	17
6. References	18
7. Acknowledgements	19

1. Introduction

Complex systems of health indicators require access to different sources, continuous update and regular maintenance. The **NeuBiro** is an open source software for population based registers for data management, statistical analysis, and assessment of indicators. It is part of a broader framework with vision to create sustainable solution for public information in healthcare in Europe and beyond.

The **NeuBiro** software is:

- Shared = Owned by a Community: Anyone can Join - to develop or use
- Open Source = Free to Modify and Use, Widely distributable
- Industry Independent = Public
- Privacy by Design = Legally Viable
- Distributed = Efficient and Sustainable
- Standardized = Evidence-based
- Risk Adjusted = Statistically Robust
- Multidimensional = Policy Relevant

1.1. Technological stack

NeuBiro's goal is to process data stream from different sources, run a sequence of statistical operations and eventually produce a structured report about health related data. It is developed with the help of several open source projects. It uses [Groovy](#) as main programming language and the [Griffon](#) framework to implement the user interface, the internal database is powered by [H2](#) and the report is generated with [Docbook](#).

It aims to be:

- fast
- modular
- have flexible configuration
- have few external dependencies (only R-stat)
- developed with modern languages and techniques
- multi-platform (runs on the JVM)

NeuBiro is not bound to a specific field of analysis; every aspect of the software can be defined with configurations and modules.

2. Getting started

This section will guide the user through the installation of the software and its use.

2.1. Main dependencies

NeuBiro requires some software already installed on the user's computer:

Component	Minimum version
<i>Java Virtual Machine</i>	1.7
<i>R</i>	3.3.x



The statistical modules are strictly dependent from the version of R. Using a different version of the R interpreter may result in unwanted behaviors.

The Java Virtual Machine can be downloaded from here <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, it is sufficient to download the JRE and not the entire JDK that also contains the development tools.

The R interpreter can be downloaded from here <https://cran.r-project.org>.

After a successful installation of all the dependencies described above it is possible to start the installation of NeuBiro.

2.2. How to install

NeuBiro is provided with a custom installer program; its purpose is to prepare all the needed components on the destination machine.

The components part of the installation package are:

1. NeuBiro executable program
2. User's guide
3. Programmer's guide
4. Sample data set
5. Sample statistical package

The installation program is multi-platform hence can be executed on both Windows and Linux; the installer for the latest version can be downloaded at the following link:

<https://github.com/eubirodnetwork/neubiro/releases>



This document describes the installation for Microsoft Windows but the steps for the other platforms are the same.

The installation procedure guides the user during all the phases of the process. Some components are optional and can be omitted but, on first installation, we suggest to run a complete install. In this way all the sample data will be installed and the system can be tested out of the box.

Under Microsoft Windows an icon on the Desktop will be created as well as an item under the Start Menu.

2.2.1. Installation steps

To start the installation it is sufficient to double click on the downloaded `jar` file. The following images describes each step of the installation process.

Language selection

In this first phase the user can choose the language used by the installer itself.



Figure 1. Language selection

Welcome screen

The welcome screen show the version of the software

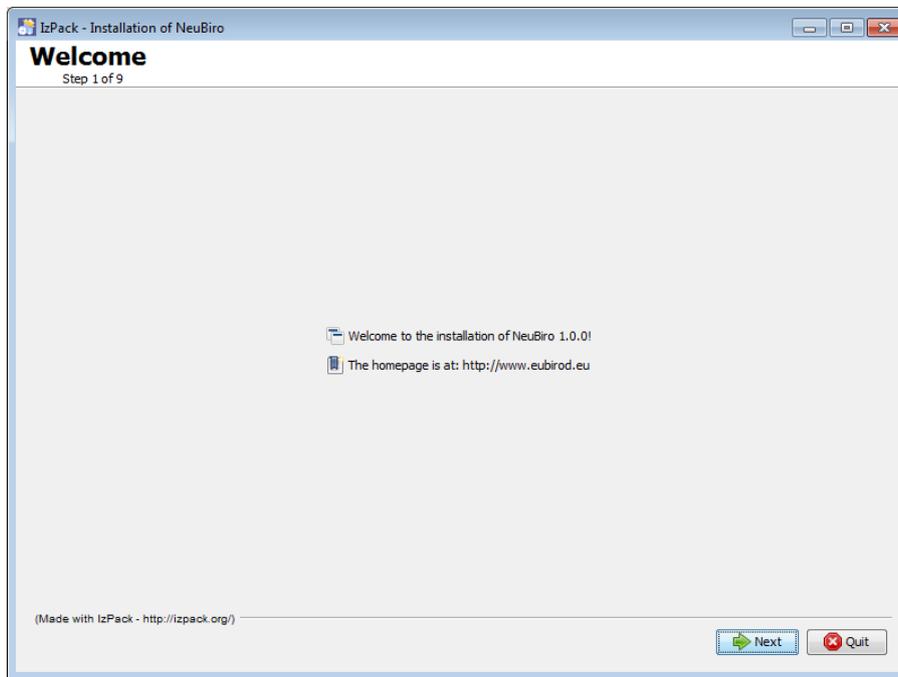


Figure 2. Welcome screen

Information about the software

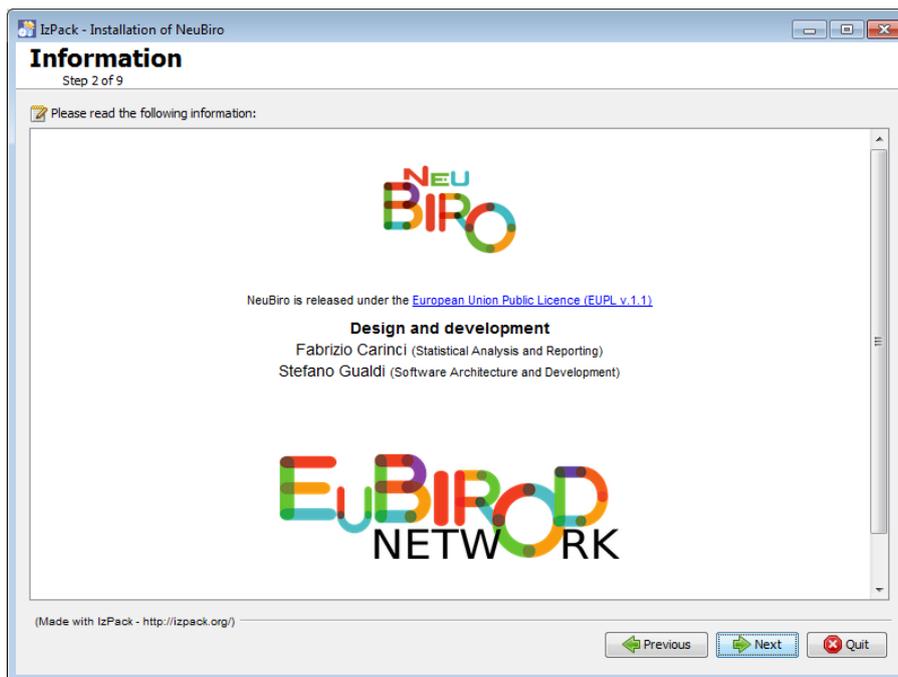


Figure 3. Information screen

License

To proceed with the installation it is required to accept the software license.

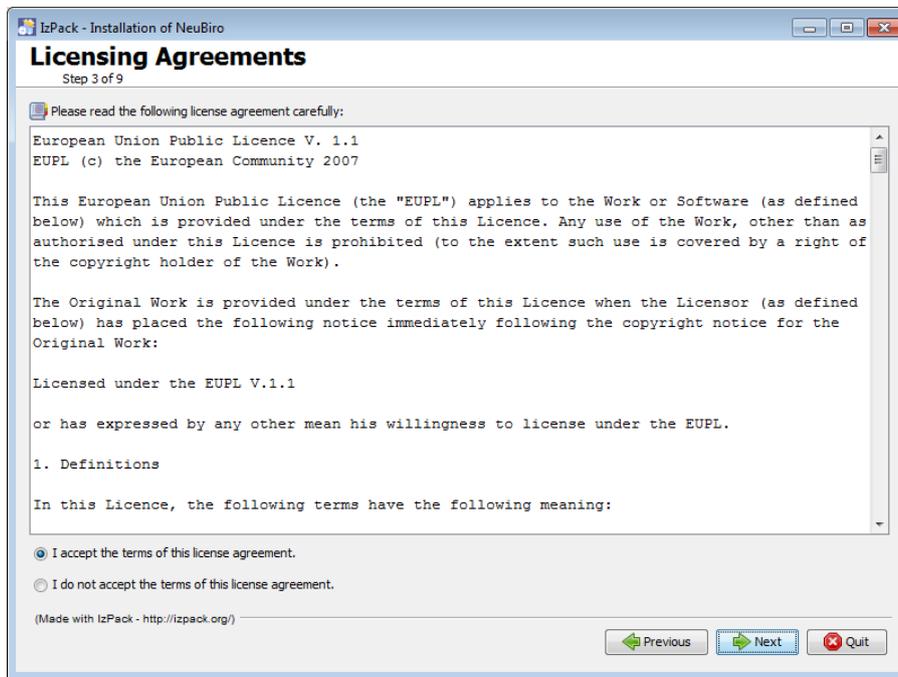


Figure 4. License

Packs selection

The installation package contains several packs, some of them are mandatory but others (like the sample data) are not and can be omitted.



On the first installation it is advisable to leave the proposed choices so that will be possible a quick test run to verify that everything is in place and all the dependencies are working.

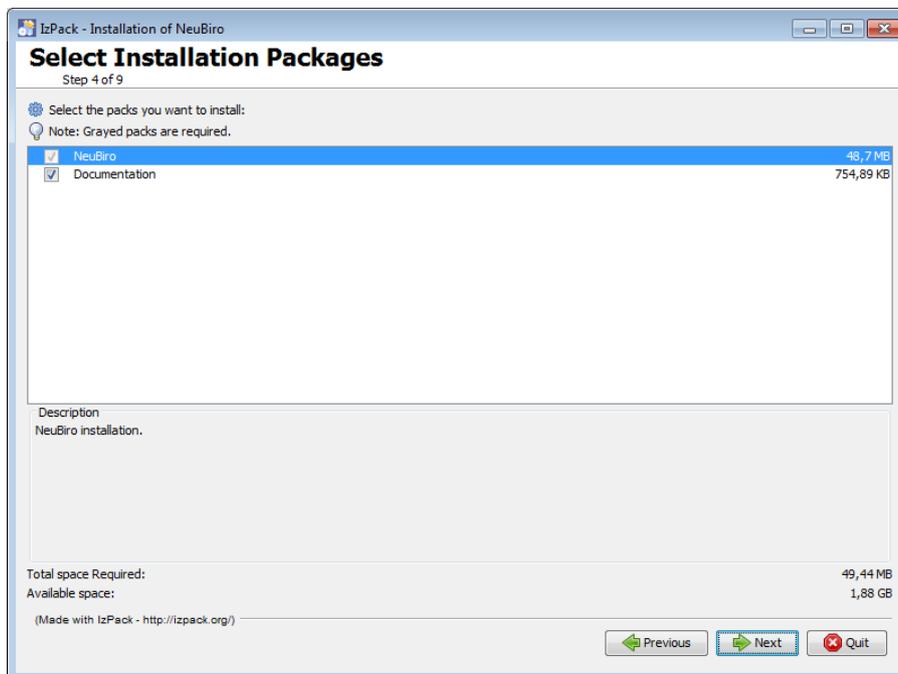


Figure 5. Available packs

Installation path

This path represents the position in which the software will reside.



It is better, at least in the first installation, to not to change the proposed destination folder. This will be useful in case of requests to the support team.

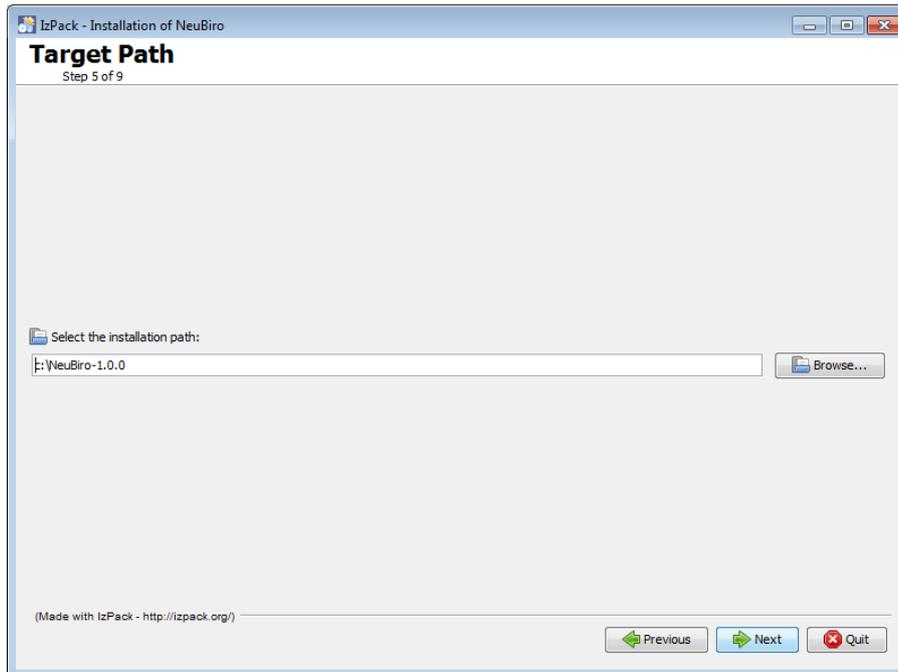


Figure 6. Installation path



If the installer complain about a non writable directory, please use a path under your home directory.

Summary

Before starting the installation a summary screen will recap the performed choices.

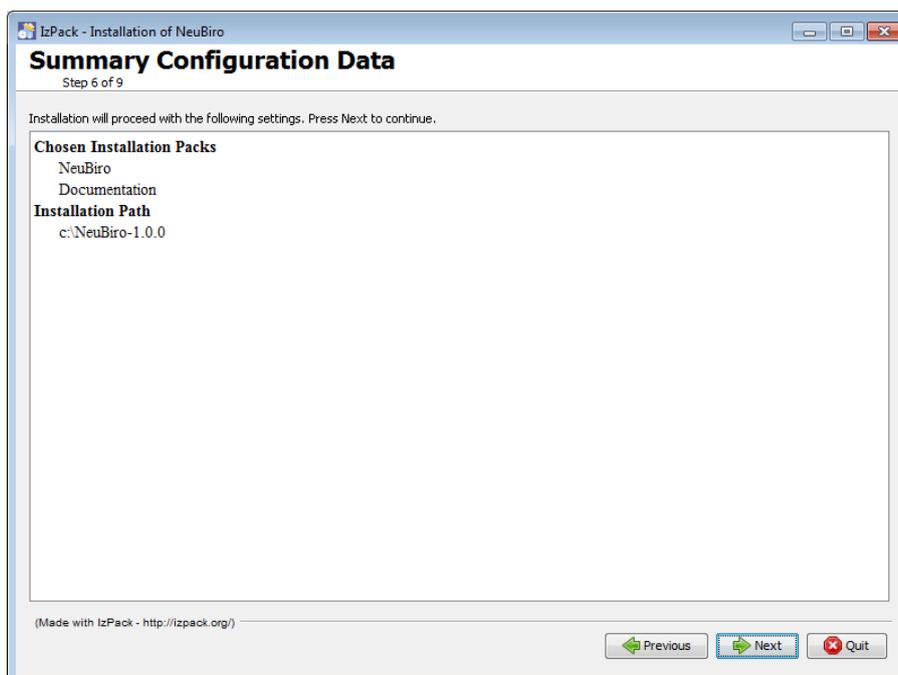


Figure 7. Installation summary

Progress

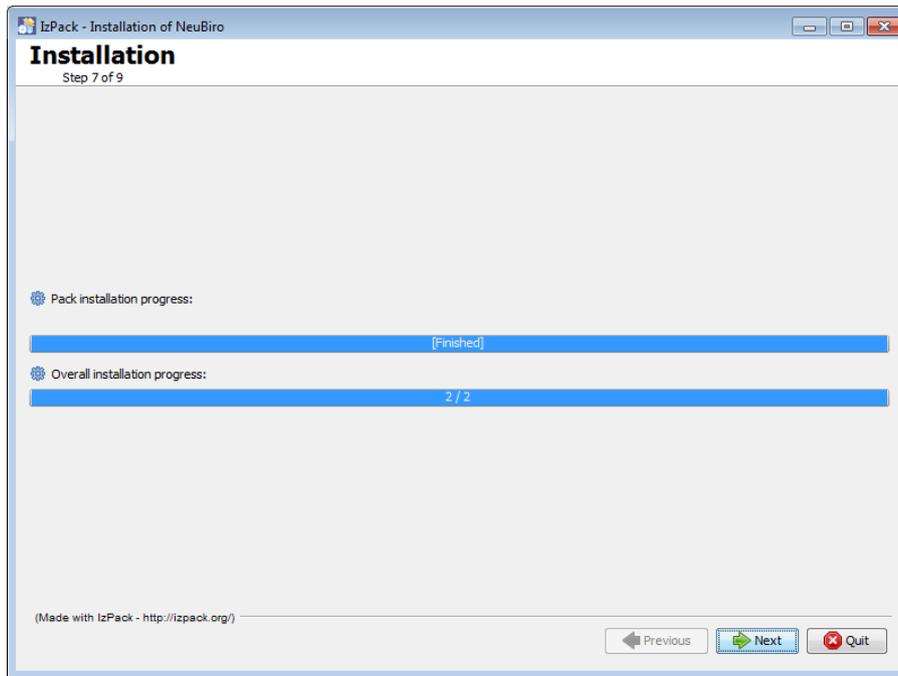


Figure 8. Progress indicator

Desktop and menu shortcuts

On this panel is it possible to change the way the shortcuts used to launch NeuBiro will be created.

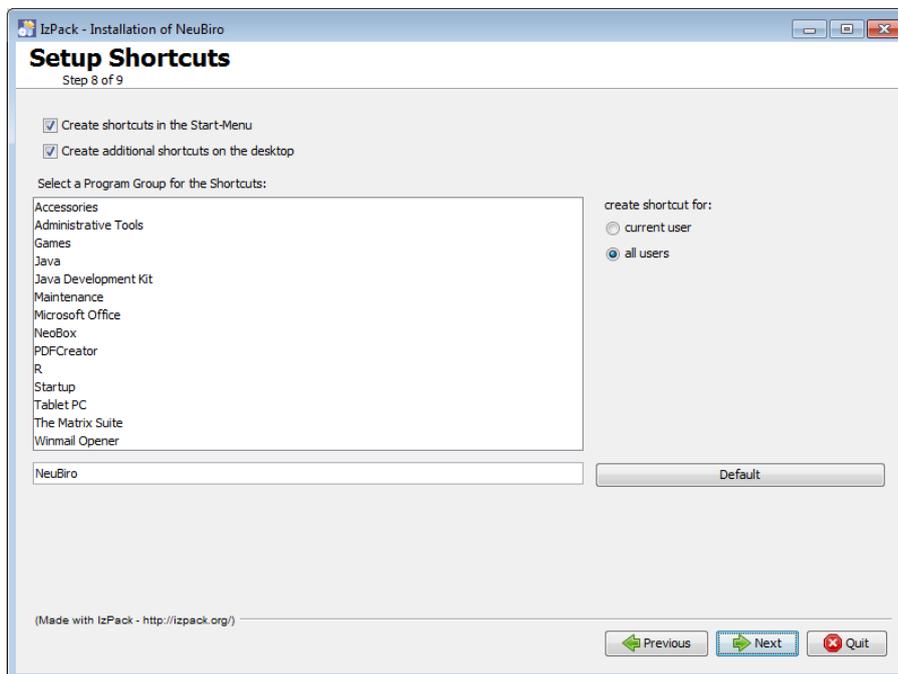


Figure 9. Windows shortcuts

Installation completed

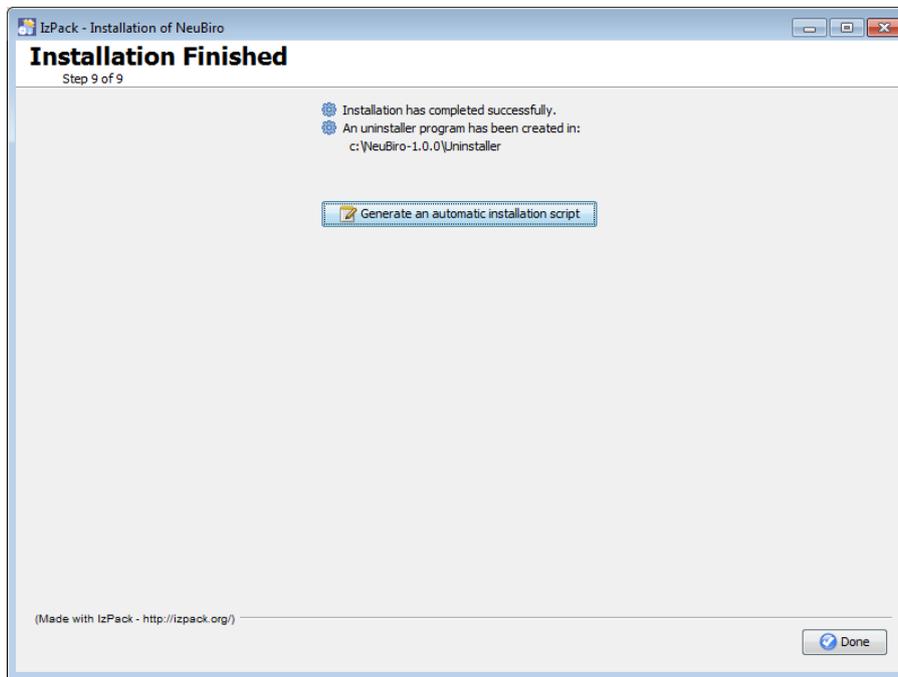


Figure 10. Installation complete

At the end of this step all the components are written on the local machine, unpacked and the user is ready to start working with NeuBiro.

2.2.2. Installation folder's layout

After the installation step all the components will reside under the folder `C:\NeuBiro-0.7`.

What follows is the layout of the folders created on the destination machine:

```
C:\NeuBiro-0.7 ①
+-- docs ②
+-- sample-data ③
`-- packages ④
    |-- sample-package ⑤
        +-- import ⑥
        |-- indicators ⑦
```

- ① Main installation folder
- ② Documentation in PDF e HTML format
- ③ Sample data set
- ④ Statistical package main folder
- ⑤ Sample statistical package
- ⑥ Import specifications
- ⑦ Statistical modules



The layout shown above is the standard one, if the user opted for different choices the root folder may be different, and some inner files or folder may not be present.

3. Working with NeuBiro

To make its work, NeuBiro needs some components that must be provided by the end user. These component can be summarized as follow:

- the master data set containing all the data to be analyzed
- the import specifications that describes the above data set
- the statistical package containing all the code needed to perform the analysis.

3.1. Overview of the working pipeline

The following images summarizes the entire pipeline of NeuBiro

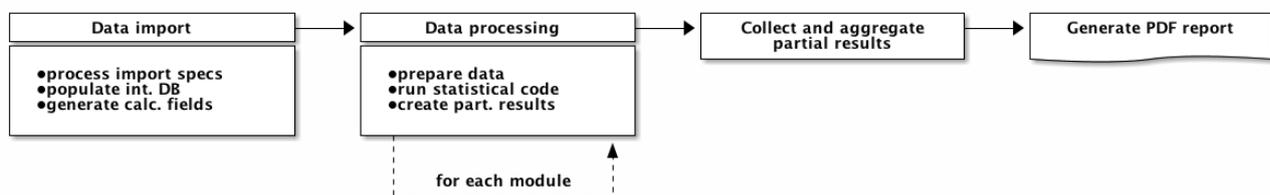


Figure 11. overview of the main workflow

In the following sections each step will be described with greater detail.

3.2. The user interface

When launched NeuBiro presents to the user a simple graphical interface.

The user interface is divided in tabs and each section represents a different step in the pipeline:

- Configuration
- Import
- Analysis
- Transfer
- Log

3.3. Configuration

The configuration screen is used to enter the basic data of the user and to select the main language of the interface. The main language, if not selected or at the first run, will be inferred by the system language defined on the host operating system. At the moment the supported languages are English and Italian.

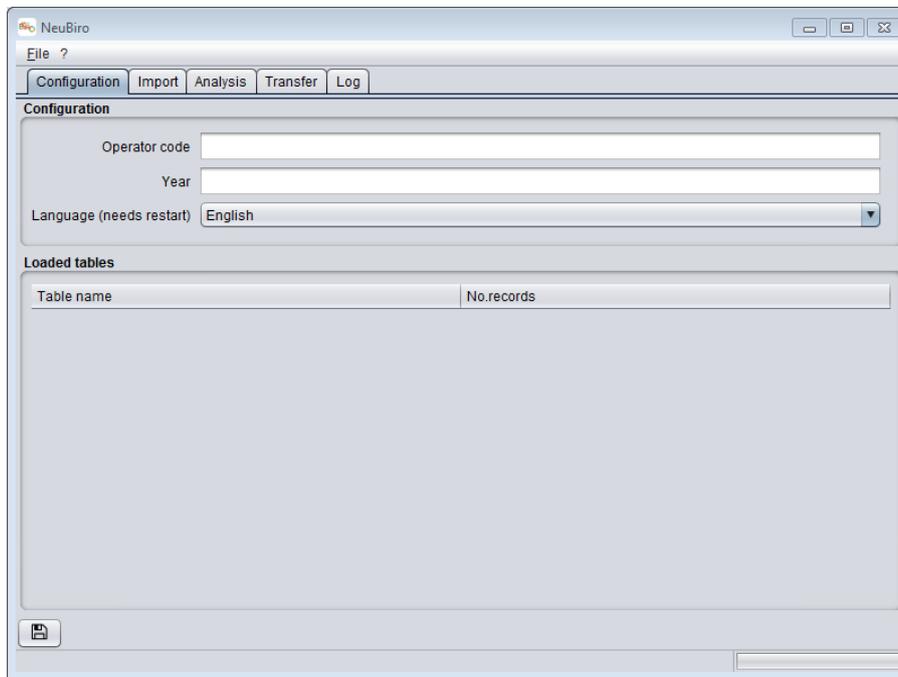


Figure 12. The configuration tab

The lower part of the screen shows all the tables already imported and the numbers of records contained in each of them.

3.4. Importing the data set

The first phase is the one in which the data are read from the data set and the internal relational database is created. This step is driven by the import specifications. Import specifications are represented by a text file containing a description of the data to import and, if necessary, the definition of new fields.

An example of an imports specification file is as follow:

```

master { ①
  'THETABLE' { ②
    context { ③
      ...
    }
    fields { ④
      ...
    }
    calculatedFields { ⑤
      ...
    }
  }
}

```

- ① master block
- ② master table block
- ③ variables definition

- ④ fields definitions
- ⑤ calculated fields definitions

The end user is not required to write this file, in fact it is usually provided together with the statistical package that we will describe later. All the information for writing your own analysis can be found on the Programmer's guide.

Thanks to this configuration, at the end of the import process, the internal database will be populated with all the data necessary to the next steps. That's the reason why the import specifications file is usually shipped with the statistical package, in fact the latter heavily relies on the names defined in it.

The import must be done before starting the analysis and can be repeated at any time in case the data changes. If the source data set does not change there is no need to perform the import again as all the data remains persistent on user's machine and will be automatically loaded.

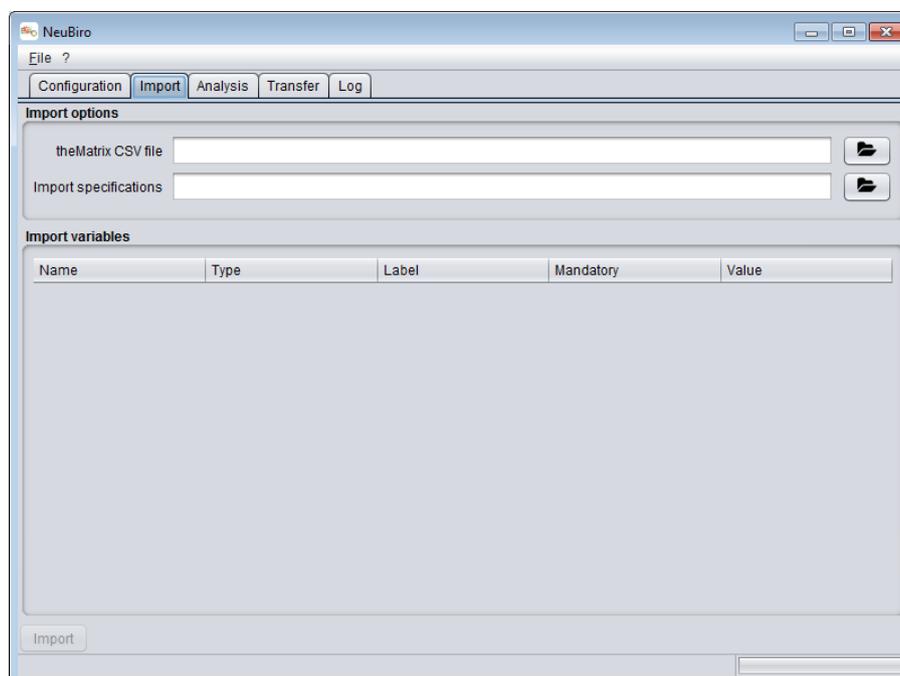


Figure 13. The import tab

The main data to be defined here are:

- the path to the data set file
- the path to the import specifications file
- optionally, if defined in the specifications, the context variables are also requested.

After filling the required fields the import process can be started by pressing the IMPORT button.

This operation can take several minutes and depends on the amount of data to be imported; during the import the panel will be dimmed and the status bar will show information about the ongoing activity.

At the end of the import all the data, as described in the specifications file, will be ready to be used for the analysis step.

3.4.1. How to prepare the data set for NeuBiro

The input that NeuBiro expects is represented by a CSV (Comma Separated File) file like the one in the following example:

```
NAME,SURNAME,BIRTH_DATE
John,Doe,15/03/1970
Mary,Doe,24/10/1970
...
Michael,Smith,10/01/1968
```

The fields' separator is , (comma).

TBW

3.5. Running the analysis

The core of the system is represented by the Analysis tab. The goal of the analysis is to run the statistical code, that in turn will access the imported data, and to produce the final pdf report.

On this screen the user can find all the options required to run the analysis phase.

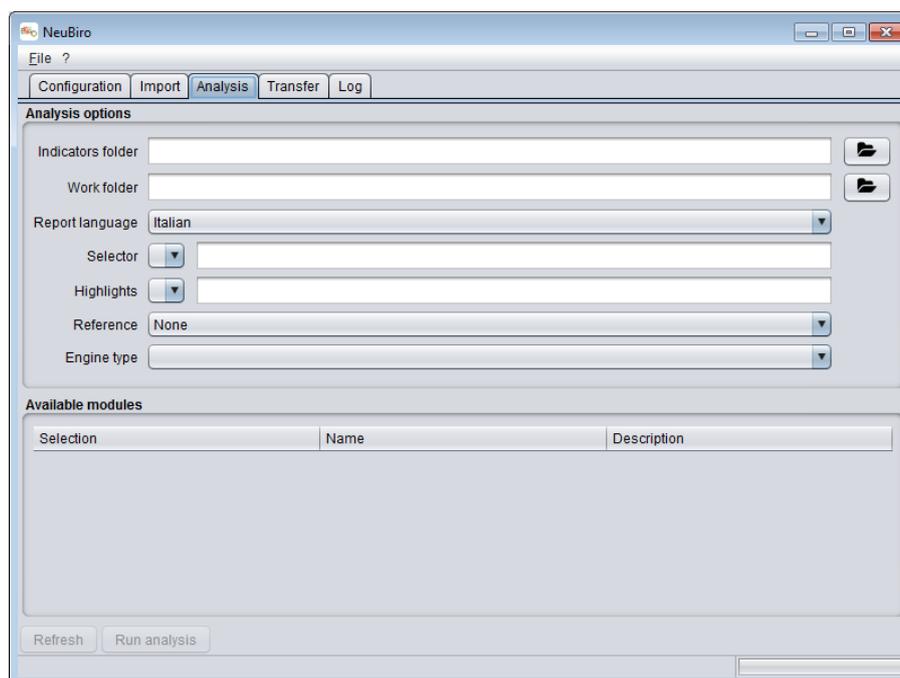


Figure 14. The analysis tab

The mandatory information are:

- the indicators folder
- the work folder

The indicator folder is the path to the directory that contains the statistical package, once selected all the modules included in the package will be listed on the second part of the screen. From the list the user can select one or more modules, each selected module will collaborate to the production of the

final report.

The [work folder](#) is the path to an empty folder in which all the partial data and the final report will be saved.



The fields presented on this screen depends on the statistical package used, hence they are not mandatory and may be left blank. Please refer to the documentation provided with the statistical package to know what options can be used.

3.5.1. Advanced analysis

TBW

3.5.2. Organization of the work folder

During the analysis several files and folders are created under the specified work directory, The basic organization of the can be summarized as follow:

```
<WORKING-ROOT>
|-- SINGLE
   |-- 1
   |-- 2
   |-- ...
   |-- N
   |-- report
```

At the end of the analysis phase, under the root folder, there will be a **SINGLE** folder containing all the artifacts produced. There will be a folder for each indicator processed and the name will be equal to the id of the indicator, and a **report** folder containing the final report generated.

These folders can be useful for debugging purposes as in each one all the output from the statistical routing will be written into a file named **indicator.log**. In the case something go wrong this file can be used to identify the problem.

3.6. Transfer the aggregated data

The transfer tab permits the upload of the aggregated data to a central server. At the moment the available protocol is FTP.

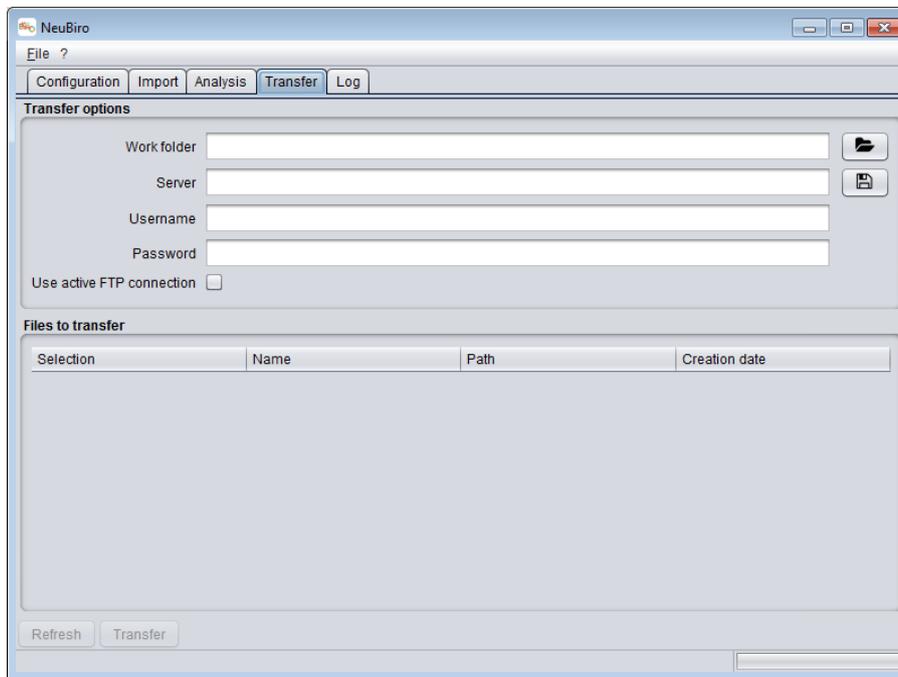


Figure 15. The transfer tab

The data that can be uploaded will be showed on the lower part of the window, but only in case the used statistical provides this function. In fact, the statistical package, usually provides a specific module that generates the zip artifacts.

Please refer to the documentation of the statistical package to know the details about this feature.

3.7. The logging tab

The logging tab shows the output of all the operation performed eg import, analysis, etc. It also reports the execution of each performed task.

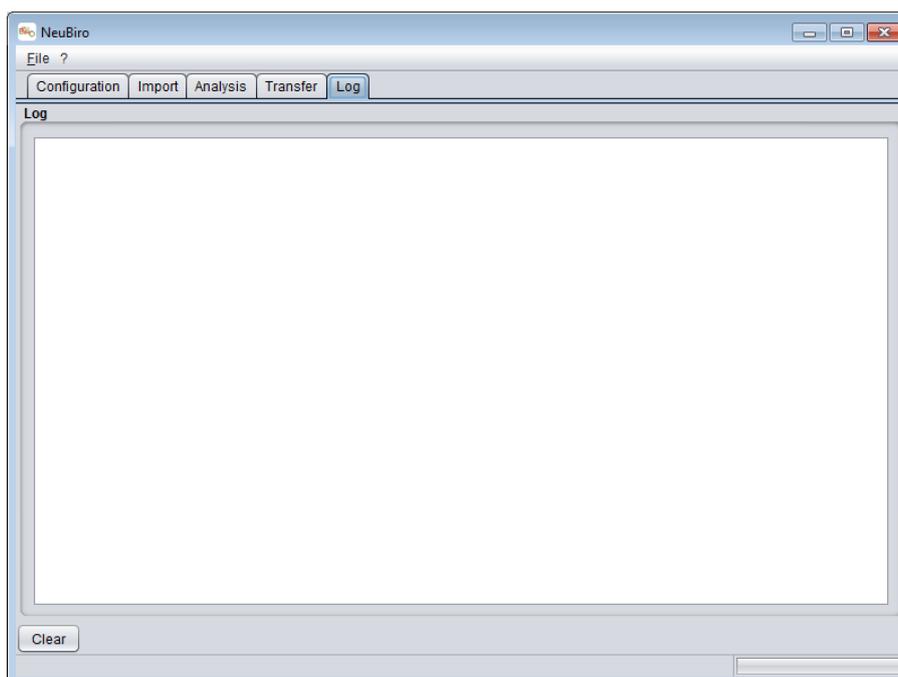


Figure 16. The log tab

4. Conclusion

In this manual we covered all the steps needed to start your analysis with NeuBiro.

If you need help please refer to the EUBIROD Network community at <http://www.eubirod.eu>

5. FAQ

The installer complain about the installation directory is not writable, how can I proceed?

The installer tries to create the installation directory on `c:\`, on some system (eg Windos 10) this folder cannot be accesses without explicit permissions.

To solve the problem change the target directory to be under your personal user folder eg `C:\Users\myname\NeuBiro-x.x.x`.

6. References

NeuBiro is build with the help of great open source projects:

Project	Description	URL
The Groovy language	a multi-faceted language for the Java platform	http://groovy-lang.org
The Griffon framework	a desktop application development platform for the JVM	http://griffon-framework.org
H2 Database Engine	embeddable relational database engine	http://www.h2database.com
DocBook	a semantic markup language for technical documentation	http://docbook.org
AsciiDoctor	a fast text processor and publishing toolchain	http://asciidoctor.org

7. Acknowledgements



This section is a placeholder, must be detailed with tasks from each contributor.

All contributions, in any form, should be reported here.

The main developers for NeuBiro are:

Developer	Role
Stefano Gualdi	NeuBiro software, documentation, Java/Groovy programming
Fabrizio Carinci	Statistical analysis and programming of the statistical packages, R Language
Iztok Stotl	Documentation, beta testing