



**EUBIROD**  
EUropean Best Information  
through Regional Outcomes in Diabetes



Executive  
Agency for  
Health and  
Consumers

## **WP5 DATA COLLECTION**

**DELIVERABLE D5.3**

**DATABASE ENGINE**

**August 2010**

*This report is Deliverable D5.3 Database Engine of “WP5: Data Collection”, the European project “European Best Information through Regional Outcomes in Diabetes” (EUBIROD), co-funded by DG-SANCO, European Commission, 2008 (G.A. 2007115)*

**Scientific Coordinator:** *Prof. Massimo Massi Benedetti*

**Technical Coordinator:** *Fabrizio Carinci*

**A joint production of the EUBIROD Consortium:**

*Adelaide and Meath Hospital, Dublin, Ireland  
Centre Hospitalier de Luxembourg, Luxembourg  
Dutch Institute for Healthcare Improvement, Netherlands  
Havelhöhe, Berlin  
Hillerød University Hospital, Hillerød, Denmark  
IMABIS Foundation, Malaga, Spain  
International Diabetes Federation, Belgium  
Inst. Scient. Santé Pub. WIV, Brussels, Belgium  
Joanneum Research, Austria  
Medical University of Silesia, Katowice, Poland  
Ministry of Health, Cyprus  
NOKLUS, Norway  
Paulescu Institute, Romania  
Sahlgrenska Academy, Gothenburg, Sweden  
Sereatrix snc, Italy  
University of Dundee, Scotland  
University of Malta, Malta  
University of Perugia, Italy  
University of Debrecen, Debrecen, Hungary  
University Children’s Hospital, Ljubljana, Slovenia  
Vuk Vrhovac University Clinic for Diabetes, Zagreb, Croatia*

**WP Leader:** *University of Perugia, Italy*

**Compiled for the University of Perugia by:**

*Valentina Baglioni, Software Engineer, EUBIROD Project Manager  
Fabrizio Carinci, Senior Statistician, EUBIROD Technical Coordinator*

**Secretariat**

*Anna Rita Ragni, Secretary, EUBIROD Coordinating Centre, University of Perugia, Italy*

**Citation**

*V. Baglioni, F. Carinci on behalf of the EUBIROD Consortium, Database Engine, EUBIROD Consortium, 2010*

**Address for correspondence**

*EUBIROD Coordination Centre  
Via E. dal Pozzo 06126 Perugia – ITALY  
Ph/Fax. +39 075 5727627  
Email: [eubirod@unipg.it](mailto:eubirod@unipg.it)*

**Project Website** <http://www.eubirod.eu>

# **Index**

<b>Executive Summary.....</b>	<b>1</b>
<b>1. Introduction.....</b>	<b>2</b>
<b>2. Objectives.....</b>	<b>3</b>
<b>3. Materials and methods.....</b>	<b>5</b>
3.1. Revised BIRO System architecture .....	5
3.2 Uniform data import approach.....	6
3.3 Synchronization with the On Line Data Source Questionnaire.....	7
3.4 Optional XML Import Export.....	8
3.5 Additional Wide XML schemas.....	9
3.6 Customized Toolbox .....	11
3.7 Data Quality Check.....	11
3.7.1 Merge Table Quality Check.....	12
3.7.2 Activity Table Quality Check .....	16
3.8 Statistical Engine Preprocessing.....	19
3.9 Embedded central engine.....	20
<b>4. Results.....</b>	<b>21</b>
4.1 Working directory.....	21
4.2 BIROBox layout.....	22
4.3 Setup panel.....	23
4.4 Configuration Validator.....	24
4.5 Configuration Editor.....	25
4.5.1 Data source profile configuration .....	26
4.5.2 Data input configuration .....	27
4.5.3. Field Mapping Configuration .....	28
4.6 Inspector.....	29
4.7 Statistical engine panel.....	30
4.8 Statistical engine browser.....	31
4.9 Central engine panel.....	32
<b>5. Discussion.....</b>	<b>33</b>
5.1 Consortium feedback.....	33
5.2 Conclusions and perspectives.....	34
<b>References.....</b>	<b>35</b>

## Executive Summary

The activity of WP5: “Data Collection” dedicated to the Database Engine was aimed at loading EUBIROD data according to the agreed format, through a user friendly interface that can be adopted by all partners on a routine basis (the “BIROBox”).

The Database Engine was designed to comply with all updated specifications of the BIRO system.

The activity included the following tasks:

- linking the PostgreSQL database to the revised EUBIROD Common Dataset
- checking the quality of the data submitted
- pre-processing data for the statistical engine
- implementing new options
- triggering functions of the statistical engine
- browsing statistical outputs
- loading statistical objects into a central database
- triggering functions of the central engine
- transferring statistical objects to the central server

The initial revision of the software was undertaken at the Rome Technical Meeting 2009. Feedback was requested from all partners. Consequently, a technical document listing all necessary improvements was drafted and agreed among members of the core development team.

The Special BIRO Academy Meeting 2010 allowed to evaluate the initial revision with all partners of the Consortium. The final improvements were agreed and subsequently implemented between June and August 2010.

This activity has provided a crucial contribution to the new version of the BIRO system.

The results included in this report document the following:

- more reliable routines to control for data quality at input
- enhanced flexibility to allocate different formats from the local data sources; increased performance and scalability to load input data of different size into the BIRO PostgreSQL database
- comprehensive interface allowing direct monitoring of the data import process and a simplified use of the statistical engine.

All software developed for the Database Engine has been included in the BIROX distribution, a cross-platform, virtualized comprehensive release of the software powered by Linux Ubuntu.

Feedback received by all partners confirm that the current release of the Database Engine embedded in the BIROBox and running on BIROX has definitely improved the possibility for the EUBIROD Consortium to manage local data and obtain statistical results. At the end of this task, the majority of partners were able to deliver complete reports on their own.

Further improvements have been made to make the process easier and more sophisticated, particularly with regards to extensive quality checks.

The final part of the work undertaken can now focus on the usage of the BIRO system for the delivery of European Diabetes Report. These operations will allow to further refine and test the usage of the system.

## 1. Introduction

EUBIROD aims to implement a sustainable European Diabetes Register through the coordination of existing national/regional frameworks and the systematic use of the BIRO technology. The system will fulfil the Conclusions of the EU Council for the systematic data collection and monitoring of diabetes complications and health outcomes across Europe.

EUBIROD targets the sustainability of complex systems of health indicators requiring continuous update and regular maintenance. The project proposes an action to implement, extend, and customize the application of the BIRO technology in at least 20 States, including EU Member States, Acceding/Candidate Countries, and EFTA Countries.

Participants are connected through a system that safely collects aggregated data and produces systematic EU reports of diabetes indicators, which can be used to develop recommendations for policy makers.

The EUBIROD Consortium includes all BIRO Partners and N=12 new Partners coming from all over the Europe and two collaborating institutions.

The extension of the BIRO system to the whole EUBIROD Consortium required significant efforts to adapt the technology to a wider set of users and different methods of data collection.

An immediate verification of the degree of coherence between data collected by participating centres and the original BIRO common dataset highlighted that several adjustments were needed and that the development of solid procedures for quality check was absolutely necessary to avoid the production of results of dubious comparability.

Consequently, the structure of the Database Engine originally specified had to be revised accordingly, also to increase the degree of efficiency both in terms of flexibility in input/output and level of overall performance for the benefit of all partners.

A series of test and training sessions were needed to identify precise aspects of the BIRO system in need of revision and subsequently develop the required routines, also including enhanced features of the graphical user interface.

The present Deliverable, D5.3, includes a detailed description of architectural changes and software enhancements applied to the database components and the interface of the BIRO system in order to meet the above requirements. The report includes a detailed presentation of the new release of the BIRO software and a complete user guide included as an appendix.

## 2. Objectives

The aim of the EUBIROD work package 5 “Data Collection” is to ensure the maximum usability to the BIRO system and to optimize its performance based on the requirements of the broad EUBIROD Consortium.

Three core BIRO work packages were submitted for revision to the entire Consortium: the Privacy Impact Assessment, the Common Dataset and the Database Engine.

All partners were asked to fill ad hoc questionnaires through an online platform allowing to collect information about data collection procedures in each region in terms of compliance to the European Privacy legislation and BIRO Data Definitions. Results of these surveys are summarized by deliverables D5.2 “EUBIROD Privacy Impact Assessment”<sup>1</sup> and D5.1 “EUBIROD Common Dataset”<sup>2</sup>.

The revision of the Database Engine started immediately after the extensive training and test session held at the first BIRO Academy meeting (Kuwait City, 2nd-4th May 2009), documented in Deliverable D4.1 “Report on Training”<sup>3</sup>. In this occasion all partners gained an opportunity to get acquainted with the BIRO system while at the same time gave developers could test strengths and weaknesses of the available software. The direct impressions of participants were collected and used as a checklist to drive the necessary improvements to facilitate usage of the software.

In particular, a special attention was dedicated to the following aspects:

- *reliability*: allowing a greater flexibility to the input dataset requires for the system to face the problem of variable levels of data quality. Available datasets within the Consortium are not homogeneous, due to the different purposes, procedures in place and the geographical coverage of data collection. Furthermore, long term longitudinal datasets may present internal inconsistencies that can be difficult to trace from the outside. The BIRO system needs to simplify the approach while ensuring the functionality of all data loading operations and the delivery of accurate data as input for the statistical routines. Therefore, original data must be “cleaned” in advance before statistical processing is undertaken. Any large deviations from an acceptable level of quality, must be flagged to the user through adequate warning messages, so that correcting actions can be undertaken through an examination of internal error logs.
- *flexibility*: the BIRO system must face the problem of local datasets showing a significantly different structure from the BIRO dataset. In many cases mapping the original data to the standard definitions is far from being straightforward. The BIRO system must provide the user with additional functionalities to bridge the available dataset with the expected shape of the BIRO database. Even slight differences can imply long execution times for the average user. Therefore, developers agreed that improved ETL functionalities (extraction, transformation, load) were absolutely necessary.
- *scalability*: the BIRO system must ensure broad functionality and an acceptable level of performance regardless of the size of input datasets. Registers involved in the EUBIROD Consortium maintain very large databased that can be difficult to handle if efficient routines are not adequately implemented. Besides, the more the network grows, the higher the probability that very large databases can pose problems of difficult resolution to the BIRO software.
- *performance*: in some cases the execution time of the BIRO system showed to be inadequate compared to the average user expectation. The internal data flow presented a long chain of cascade processing. Some of the steps turned out to be a bottleneck in the chain since they were time-consuming and required lot of memory. The internal structure of the database was perceived to be unnecessarily complex for the user needs, resulting in long waits for data loading and database processing even for datasets of average size.

- *user-friendliness*: users asked for a simplified interface to monitor data import and statistical processing, without interacting at a low level with the PostgreSQL database, which could be potentially dangerous for the integrity of the database if handled by users without the required level of technical skills.

Each goal described above was translated into a specific task for the activity related to the EUBIROD database engine.

The following objectives were targeted by activity 5.3:

- *reliability*: to develop extensive routines for data quality check and add them to the database engine to filter all data inconsistencies before handling all data to the statistical engine. Results of the quality check must be summarized into a small report containing a quality evaluation and the detailed list of problems encountered. Such a report shall be used for a self-evaluation made by the user in terms of proper data collection methods, data accuracy, and correct usage of the BIRO system.
- *flexibility*: to revise the BIRO system architecture to make possible for the user to execute complex transformation from the local dataset to the BIRO standard. The BIRO system shall act as a platform where customized ad-hoc transformations, linked to a customized toolbox, can be easily mounted and executed as deemed necessary.
- *scalability*: to optimize the access to the DBMS by restructuring routines to insert and retrieve data from the BIRO database.
- *performance*: to revise the original architecture of the BIRO Database Engine to optimize the internal data flow, to speed up the data import procedure, and to facilitate the tasks assigned to the statistical engine; to remove bottlenecks in the import process and make some tasks (e.g. XML export) optional; to reduce the burden of bulk data preprocessing by moving it from the statistical engine to the database engine to make the process more efficient at a lower level.
- *user-friendliness*: to add inspection tools to the user interface in order to allow users: a) to check the input dataset while configuring mapping of local fields to the BIRO format; b) to compare the output of the import process to the original dataset; c) to examine the data quality check through a simple statistical report; and d) to easily browse outputs produced by the statistical engine.

The aim of the present document is to detail all the technical updates realized for the the BIRO system to meet the above requirements.

### 3. Materials and methods

#### 3.1. Revised BIRO System architecture

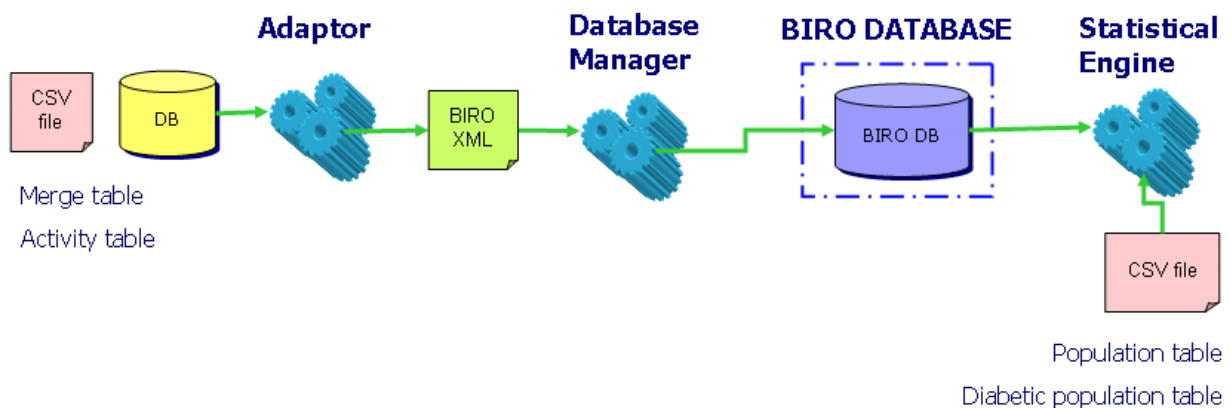
In this section we compare the original BIRO system architecture with the revised one and explain in detail changes applied to improve the following five quality factors: reliability, flexibility, scalability, performance, user-friendliness.

Diagrams included in Figure 1-2 describe the BIRO System architecture before and after the revision.

The initial BIRO System architecture presented a very simple linear structure<sup>4,5</sup>.

Data processing included the following three major steps:

- The “Adaptor” reads local data (clinical patient data and activity data) from database tables or CSV files, maps them to the BIRO standard format and produces a set of BIRO XML files representing each patient separately.
- The “Database Manager” reads the BIRO XML files and imports them into the BIRO database
- The “Statistical Engine” extracts clinical patient and activity data from the BIRO database and produces statistical indicators directly from CSV files; produces HTML and PDF statistical reports and calculates statistical objects to be sent to the Central Engine for the global report.



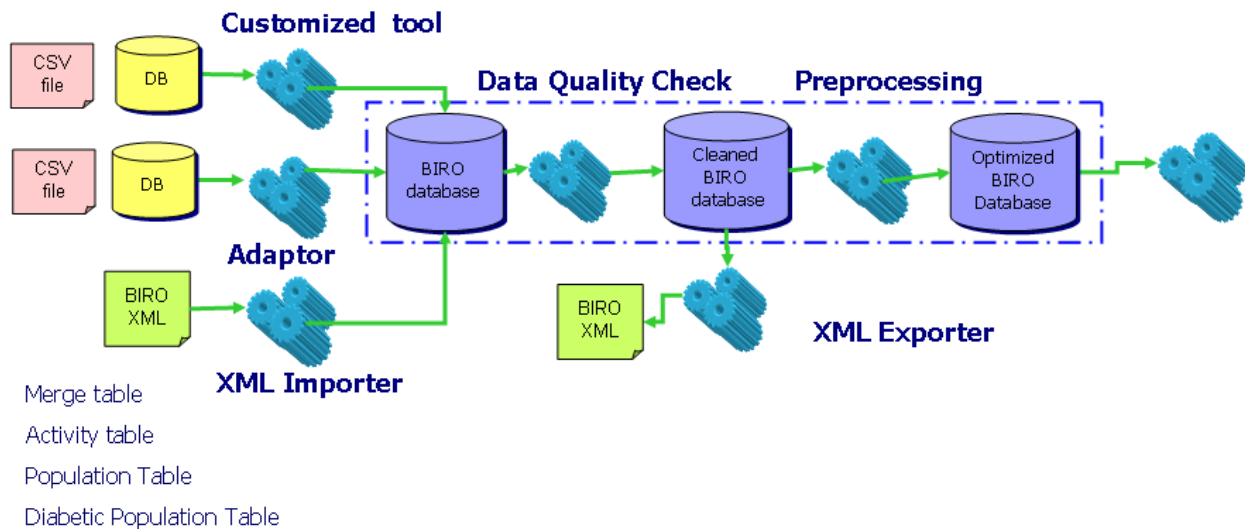
**Figure 1: BIRO System architecture before revision**

The revised version of the BIRO System architecture shows an increased architectural complexity. Some tasks have been reshaped and more steps have been included in the data processing chain:

- Adaptor and XML Importer allow the system to read clinical patient data and activity data as well as population and diabetic population data from XML files, CSV files or database tables
- data can be loaded directly into the BIRO database without any intermediate step
- optionally the system supports customized transformations allowing complex mapping strategies between local dataset and the BIRO format
- a data quality check has been applied to the BIRO Database. The filter detects unacceptable values as well as inconsistencies among values within the same record, two different records or even two different tables



- optionally the cleaned BIRO database can be exported as XML files
- the statistical engine reads optimized tables from the BIRO database, specifically created by a preprocessor to speed up all statistical calculations
- as in the first BIRO system version, the statistical engine prints HTML and PDF reports and calculates statistical objects to be sent to the Central Engine for the global report



**Figure 2: BIRO System architecture after revision**

Despite of the increased architectural complexity, the new release of the BIRO system is substantially more efficient than the previous ones. The following paragraphs address the changes in detail, highlighting the practical advantages of each solution implemented.

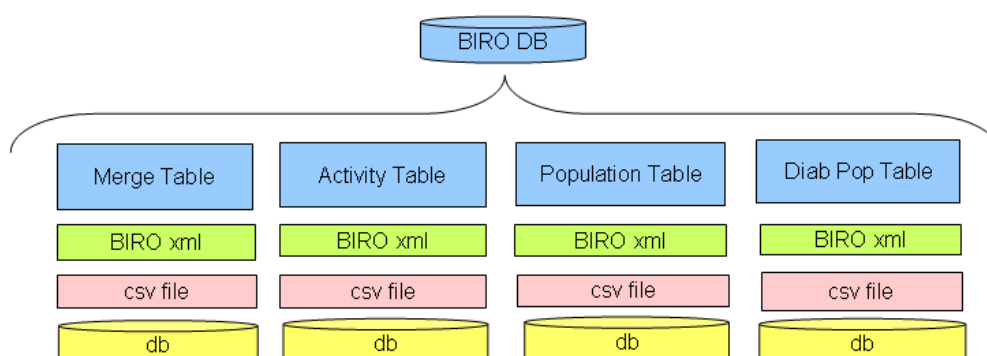
### 3.2 Uniform data import approach

The new BIRO System applies a consistent data import approach to all input data sets. Before the revision, the Merge Table and the Activity Table were submitted to the standard import process (transformation and storage into the BIRO PostgreSQL Database), while the Population Table and the Diabetic Population Table could be provided directly as input to the Statistical Engine as CSV files.

The statistical engine had to refer to two different input sources: the BIRO database and the external files. While tables in the BIRO database were submitted to an intrinsic quality check during the mapping process, the contents of the population table and diabetic population table were not filtered by the system, with an uncertain result in terms of the quality and accuracy of the files. This aspect could potentially jeopardize the application of the statistical engine, since any error in the population files could compromise the reliability of several indicators, or even break the process abruptly.

In the revised architecture, the Population Table and the Diabetic Population Table are treated exactly the same way as the Merge Table or the Activity Table. They can be provided not only as CSV files, but also as XML files or database table; they follow the same data flow (mapping, import, validation) and are finally stored into the BIRO Database, from where they can be accessed by the statistical engine.

In the revised version, the BIRO Database represents the only input for the statistical engine, making the detection of potential threats for statistical processing substantially easier.



**Figure 3: BIRO System Input**

Changing the approach to data import made possible to make same improvements in the Common Dataset and Data Dictionary. New BIRO fields have been coded in order to describe the information in the Population Table and Diabetic Population Table, and two more sections were added to the BIRO XML schema.

### 3.3 Synchronization with the On Line Data Source Questionnaire

The BIRO System can be synchronized with the On Line Data Source Questionnaire<sup>2</sup> (available at <http://questionnaire.eubiroad.eu/>).

The purpose of the on line questionnaire is to collect metadata, including: information about data sources, procedures for data collection across participant regions, quality of the local dataset. Part of the “data source profile” (Figure 4) captured by the on line questionnaire is also required by the BIRO system in order to calculate local health system indicators. Although the BIRO system does not necessarily depend from the Online Questionnaire, it can be usefully synchronized with it through the questionnaire output file.

Once the user has completed the Online Data Source Questionnaire, the whole “data source profile” can be produced as a BIRO XML file. Similarly to the Merge Table, the Activity Table and the Population Tables, the BIRO System can read such an XML file to populate the internal “Data Source Profile” with data directly fed by the online platform. The synchronization allows the user to compile the profile form only once and to avoid possible discrepancies.

At the moment this operation can only be done manually. The user must access his/her own account on the Online Data Source Questionnaire and shall download the XML file containing the profile. A future improvement may consist in an automatic download of this file.



**BIRO Academy** **EUBIROD**

Welcome Scotland [LOGOUT](#)

[Questionnaire](#) [P.I.A.](#) [Data Manager](#) [Table Manager](#) [Admin](#) [User Guide \(PDF\)](#)

User Info Site Header Site Profile Clinical Data

**Site Header**

Please use the "Save" button to persist data into the database.

Address 1 Tayside Diabetes Managed Clinical Network

Address 2 Strathmore Diabetes Centre

Address 3 Level 7, Ninewells Hospital

Address 4 Dundee

Post Code DD1 9SY

Country Scotland

Website <http://www.diabetes-healthnet.ac.uk>

Clinical representative Graham Leese Clinical E-mail [graham.leese@nhs.net](mailto:graham.leese@nhs.net)

Technical representative Scott Cunningham Technical E-mail [scott.cunningham@nhs.net](mailto:scott.cunningham@nhs.net)

Comments

[Save](#) [Cancel](#)

Figure 4: Online Data Source Questionnaire

### 3.4 Optional XML Import Export

The first release of the BIRO system directly transformed local data into the standard BIRO XML format to later import it into the BIRO database. Such a double step could not be skipped, unless the user already provided data in the BIRO XML format. In that case the process could be started directly from the Database Manager.

We have clearly identified this step as a bottleneck for the BIRO data flow due to the long execution times caused by the export-import particularly when the input local dataset is large. Moreover, not all the users seem to be interested in having their local data exported to the XML format.

In the revised BIRO system, we have separated and reshaped the import/export procedures:

- the Adaptor now reads local datasets (in form of CSV or database tables) and performs a one-shot import which populates the tables in the BIRO Database directly
- optionally, data can be imported from BIRO XML files by the XML Importer, which has the same functionality of the previous Database Manager
- optionally, after the quality check, data can be exported into BIRO XML by the XML Exporter, which recovers some of the previous functions performed by the Adaptor

In summary, the revised BIRO system does not lose any functionality of the previous release: it is still possible to import/export data from/to XML, although this is not compulsory.

XML import-export has been even improved:

- in terms of performance, because one shot import means faster import
- in terms of completeness, because the user can import/export not only patient data but the whole BIRO database
- in terms of quality, because data can be exported only after a positive quality check

### 3.5 Additional Wide XML schemas

The most relevant bottlenecks affecting the performance and the scalability of the BIRO system were related to the specific architecture of the database. In the previous version, there was a significant difference between the way imported data were stored into the BIRO database, and how data were handled by the statistical engine routines.

The database structure is described in Figure 5.

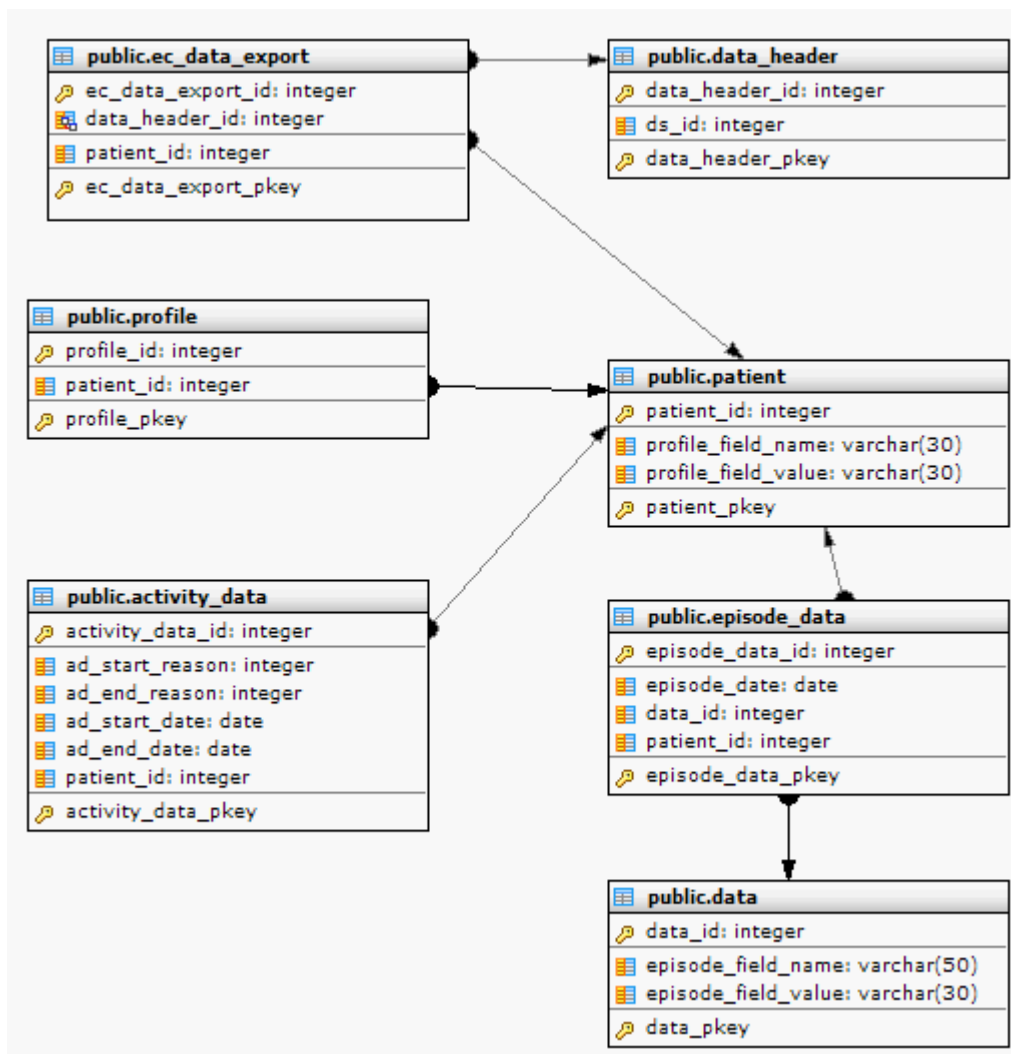


Figure 5: BIRO Database narrow format

The above structure was defined as “narrow format”, since episode and profile data were stored

in a single table with only two columns: BIROFieldName and BIROFieldValue. The approach was very neat as a gold standard for data warehouses: such structure is very powerful because it allows to be extended in order to describe complex linked hierarchical data. Because of this, it is also implemented by large pharmaceutical companies and regional governments where routine data are held and constantly expanded.

However, the “narrow format” is also more difficult to be handled for statistical analysis and programming languages like R may require substantial tweaking, with an overload on performance, to operate on top of these data. The “wide format” is better suited for the scope: here we define as “wide” a table structure containing a column for each BIRO field, similarly to the Merge Table. A “wide format “ (Figure 6) is characterized by a high number of columns, making the table containing episodes/profile data visually wider than the standard “narrow” format.

public.episode_wide	public.profile_wide
ds_id: bigint	ds_id: bigint
record_id: integer	record_id: integer
patient_id: varchar(255)	patient_id: varchar(255)
alcohol: bigint	dob: date
alc_stat: varchar(255)	dt_diag: date
amput: varchar(255)	pat_id: varchar(255)
antiplatelet_therapy: varchar(255)	sex: varchar(255)
injections: real	type_dm: varchar(255)
blind: varchar(255)	
bmi: real	
cigs_day: bigint	
creat: bigint	
education: varchar(255)	
dbp: bigint	
esrf: varchar(255)	
epi_date: date	
retinal_exam: varchar(255)	
foot_exam: varchar(255)	
pulses: varchar(255)	
ftsens: varchar(255)	
ulcer: varchar(255)	
hba1c: real	
hdl: real	
height: real	
hypertension: varchar(255)	
hypert_med: varchar(255)	
drug_therapy: varchar(255)	
laser: varchar(255)	
ldl: real	
lipid_therapy: varchar(255)	
macula: varchar(255)	
ma_test: varchar(255)	
mi: varchar(255)	
pump_therapy: varchar(255)	
retina: varchar(255)	
self_mon: varchar(255)	
smok_stat: varchar(255)	
stroke: varchar(255)	
sbp: bigint	
chol: real	
tg: real	
weight: real	
durvisit: real	
agevisit: real	

**Figure 6: BIRO Database wide format**

Before the revision, the 'narrow' format had to be translated into 'wide' format by the Database Engine. A series of complex SQL queries have been defined to create and populate correct 'wide' tables. Unfortunately, as the data size increases, the efficiency of such transformation decreases very quickly. Therefore, such operation turned out to be a bottleneck both for performance and scalability of the system.

In order to solve this problem we decided to:

- expand the BIRO XML schema throughout the implementation of additional wide sections mirroring the wide table structure
- implement new Database Manager functionalities allowing a direct, faster import from new "wide" XML files
- implement an additional Adaptor function allowing to store local data directly into the wide schema, without the "narrow format" intermediate step.

### 3.6 Customized Toolbox

The well known Adaptor mapping strategy can be only used when the local dataset is very similar to the BIRO format, but it is not suitable when the local dataset needs more complex transformations. What if we need to join several tables or merge columns? In order to answer this question, we added a new feature to the BIRO System: the customized toolbox<sup>6</sup>.

The Customized Toolbox wraps up an open source ETL (Extract, Transform, Load) tool named Pentaho Kettle<sup>7</sup>. Kettle has been specifically designed as a data integration tool that can efficiently translate a data structure into another, even when this requires many difficult transformations.



**Figure 7: Customized Toolbox**

A set of Pentaho transformations can be described through a single XML file. When a transformation file is available, the BIRO system triggers the Pentaho engine executing the transformation. The customized toolbox acts as a black box. Users do not need to perform any special operations. Such a procedure can be undertaken by partners who would prefer to skip the manual configuration, regardless of the local dataset complexity.

The XML Transformation files can be generated by users through the Pentaho graphical user interface (Spoon) or they can be provided directly through the activities included in WP7 under the leadership of Joanneum Research.

### 3.7 Data Quality Check

In the previous release, the BIRO system had no specific data quality check. The "Adaptor" only performed detection and removal of unparseable values, i.e. values with unexpected format. At the end of mapping, the list of unparseable values was displayed in the log window.

The data quality check turned out to be an indispensable task for BIRO in general, for the

following reasons:

- it reduces the risk of failures or unexpected behaviors during statistical processing
- it reduces the risk of biased results in the statistical indicators
- it helps highlighting mapping errors (selection of wrong units of measurement for numeric fields, wrong values for enumerated fields, wrong format for date fields)
- it makes the user aware of any potential pitfall in the local data collection and variable transformation

In the current version of the BIRO System, a data quality check is performed on all data inputs (merge table, activity table, population table, diabetic population table) with a varying degree of detail.

The system systematically checks for the following:

- wrong values (wrong format, out of ranges)
- inconsistencies between values within the same record
- inconsistencies between values on different records
- inconsistencies between values on different tables

Further controls may be added in the future.

Before starting any check, the BIRO System adds a sequential identifier to each record, allowing to keep track of any modification to the original dataset. Statistics and error logs are printed into a Data Quality Check Log file for inspection.

The following sections present the details of all quality checks implemented in BIRO to inspect the quality of the datasets provided by the user.

### 3.7.1 Merge Table Quality Check

#### ***First check: data format***

Firstly, the BIRO System scans the whole dataset value by value. It detects the following cases:

##### 1. missing values

These cannot be considered as errors. However, counting missing values can be extremely useful in order to evaluate the availability of valid for each field and the whole dataset, potentially highlighting any error at other stages

##### 2. values with wrong format (not parsable).

The BIRO System expects that all the values are compliant with the format configured by the user. When parsing values, the BIRO System detects any deviations from predefined rules:

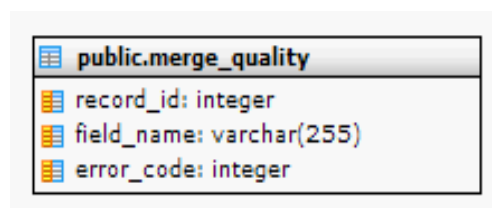
- for date fields, any string that is not compliant with the provided date pattern
- for enumerated fields any string different from those provided by mapping
- for numeric fields, any value not recognized as a number (e.g. it contains alphanumeric characters)

##### 3. values out of range

A numeric value or a date value is out of range when is greater or lower than the thresholds fixed in the Common Dataset for the corresponding field.

The BIRO System replaces values with wrong format and out of range values with null values. Missing values, values with wrong format and out of range values are then logged

into a special table named “merge\_quality”, with the following structure:



public.merge_quality	
record_id	integer
field_name	varchar(255)
error_code	integer

**Figure 8: Merge Quality Table**

where:

- record\_id is the sequential id number assigned to each record,
- field\_name contains the column name (BIRO name)
- error\_code contains one of the codes defined above (1= missing, 2= wrong format, 3=out of range).

The ranges for numeric and date BIRO fields are listed in the following table<sup>2</sup>:

BIRO code	BIRO field	Lower Boundary	Upper Boundary	Unit
BIRO010	Alcohol intake	0	5000	g/week
BIRO043	Average injections	0	20	
BIRO009	Cigarettes per days	0	100	
BIRO017	Creatinine	3	1999	umol/l
BIRO005	LDL	0.777	7.77	mmol/L
BIRO046	HDL	0.01	15	mmol/L
BIRO019	Total cholesterol	0.01	50	mmol/L
BIRO021	Triglyceride	0.259	25.9	mmol/L
BIRO015	Diastolic blood pressure	10	300	mmhg
BIRO014	Systolic blood pressure	10	400	mmhg
BIRO016	Hba1c	2.15	25.02	%
BIRO012	Height	0.3	3	m
BIRO011	Weight	5	300	kg
BIRO013	BMI	0.01	100	
BIRO007	Episode date	1900-01-01	Current Date	
BIRO005	Date of birth	1900-01-01	Current Date	
BIRO006	Date of diagnosis	1900-01-01	Current Date	



**Second check: coherence**

This check involves a comparison between values from the same record. It detects groups of values that, albeit well formatted and within their acceptability range, are not reciprocally coherent.

Currently, the following checks between date fields have been implemented:

- *date of birth vs date of diagnosis*: if the date of birth is later than the date of diagnosis, then the date of diagnosis is set to null
- *date of birth vs episode date*: if the date of birth is later than the episode date, then they are both set to null
- *episode date vs date of diagnosis*: if the date of diagnosis is later than the episode date then the date of diagnosis is set to null

Errors are logged into the same “merge\_quality” table used for the first quality check with a different error code (4).

More complex checks of coherence may be implemented in the future across multiple variables (e.g. height, weight, bmi).

**Third check: duplicates**

The statistical Engine cannot handle duplicate records. Hence they have to be carefully detected and removed from the table.

Two records from the merge table can be defined as “duplicates” when they have the same patient id and episode date. In other words, the merge table contains duplicates if there are at least two records breaking the uniqueness of the primary key.

Two approaches are possible to overcome the problem of duplicates:

- *deletion*: the dataset is sequentially scanned and every time two duplicate records are found, only one is retained (normally the last one), while the other is discarded
- *merge*: the dataset is sequentially scanned and every time two duplicate records are found a new record is created by merging the information contained in both of them. Both duplicates are discarded and only the newly created is retained.

Although the first approach would be the easiest to implement, it implies a great loss of information, so in BIRO we opted for the second option.

Whenever two duplicate records contain two non null values in the same column (same BIRO field), such values are defined as “overlapping values”.

Overlapping values are treated differently according to the corresponding BIRO field type.

For episode fields, as we assume that a patient cannot have two repeated measures on the same episode in a day, we always consider overlapping values as errors. Since the patient profile is naturally replicated over all patient records, overlapping profile values are not considered as errors, unless they are different. Profile and episode duplicates need to be handled differently. Therefore, before starting the duplicate quality check, we split the Merge Table in two separate tables: one for profiles and another for episodes.

Different strategies can be applied to manage overlapping values in duplicate records:

- retain only one value (typically the one contained in the last record) and discard the other
- calculate and retain the average value (e.g. for numeric fields)

The BIRO System has been designed to manage different merging strategies. By default, only the last value is retained.

In case of duplicate complementary records, i.e. duplicate records with no overlapping values, the merge strategy preserves all information contained.

The system stores in a table called “episode\_count” the total number of duplicates record found in the merge table for each couple (patient\_id, episode\_date). Then, it selects duplicate records from the merge table and copy them into a separate table called “episode\_duplicate” (see Figure 9).

public.episode_duplicate	public.episode_count
duplicate_id: integer	duplicate_id: serial
ds_id: bigint	patient_id: varchar(255)
record_id: integer	epi_date: date
patient_id: varchar(255)	duplicate_count: bigint
alcohol: bigint	episode_count_pkey
alc_stat: varchar(255)	
amput: varchar(255)	
antiplatelet_therapy: varchar(2...	
injections: real	
blind: varchar(255)	
bmi: real	
cigs_day: bigint	
creat: bigint	
education: varchar(255)	
dbp: bigint	
esrf: varchar(255)	
epi_date: date	
retinal_exam: varchar(255)	
foot_exam: varchar(255)	

**Figure 9: Duplicate check:  
episode\_duplicate and episode count tables**

#### ***Fourth check: cleaning***

All wrong values detected during the previous checks (values with wrong format, out of range and incoherent values) are always replaced with null values in the dataset.

Null values cannot be accepted for the key fields (patient id, episode date) and in general for all mandatory fields (patient id, episode date, sex, date of diagnosis, date of birth, type of diabetes).

To avoid that incomplete records could affect procedures of the statistical engine, we added a further step in the data quality check: “dataset cleaning”.

In this step, all records showing a null value in at least one mandatory field are discarded.

Deleted records are listed into the data quality log file as “non admissible”, so that the user may separately examine the input dataset and understand how deleted records affect the compilation of the BIRO report.

#### ***Merge table quality check report***

After the quality check, the BIRO System logs all the results into a text file. The quality check report provides the user with an overview of the quality of the local data.

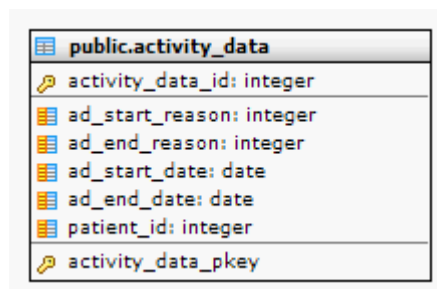
The log file includes the following items:

- Total number of :
  - missing values (absolute value and percentage)
  - values with wrong format (absolute value and percentage)
  - out of range values (absolute value and percentage)
  - incoherent values (absolute value and percentage)
  - duplicate records

- records with non admissible values in any required field
- Distribution of:
  - missing values over each field (absolute value and percentage)
  - values with wrong format over each field (absolute value and percentage)
  - out of range values over each field (absolute value and percentage)
- Detailed list of errors:
  - values with wrong format
  - out of range values
  - incoherent values
  - duplicate episode records
  - overlapping cells in duplicate episode record
  - records with non admissible values in any required field

### 3.7.2 Activity Table Quality Check

The activity table contains the patient history in the specific Data Source. It records the dates and reasons for which patients enter or leave the registry at the Data Source. Possible “entering reasons” are: birth, diagnosis and transfer from another Data Source. Possible “leaving reasons” are: death, loss-to-follow-up, transfer toward another Data Source. The structure of the Activity Table is presented in Figure 10.



public.activity_data	
activity_data_id	integer
ad_start_reason	integer
ad_end_reason	integer
ad_start_date	date
ad_end_date	date
patient_id	integer
activity_data_pkey	

**Figure 10: Activity table**

Each record describes the beginning and end of the period during which the Data Source was in charge of the individual patient. For each subject there may be multiple records in the table. Since records are linked to the merge table, the quality check must take into consideration different potential situations arising in real life conditions.

The basic idea of the activity table quality check is to create a longitudinal history of the patient encounters with the Data Source, in the form of a chronological list of events related to:

- episodes included in the merge table
- start events included in the activity table (birth, diagnosis, transfer)
- end events included in the activity table (transfer, death, loss to follow up)

The list of event is then parsed to:

- detect and correct wrong patterns (e.g.: an episode following the death event, two consecutive start events without any end episode in between,...)
- detect and correct missing start, end events (e.g. potential loss to follow up when the lag

time between two episodes is excessively large)

Finally, the activity table is reconstructed based on the revised patient history.

Such algorithm may be usefully applied in different situations, as it allows to correct and complete the activity table (when available) as well as creating an activity table from scratch when the user cannot provide it.

All the details about the algorithm may be found in the next sections.

### ***First check: data format***

The first quality check is performed on the data format as for the Merge Table. The procedure is identical: the system searches for null values, values with wrong format and out of range values. The only difference is that data format errors are then logged into a dedicated table called "activity\_quality".

### ***Second check: creating the patient diary***

The main objective of this task is to create a diary for each patient, based on the information included in both the activity table and the merge table.

We adopted the following procedure:

- split the activity table into two separate tables containing start events and end events.
- correct events with known date but unknown reason, by assigning them a fake start or end reason. Since codes ranging from 1 to 3 are already used to map possible reasons, we use the code "4" to imply a generic start or end reason.
- merge both type of events into a single table with the following structure: Event(patient\_id, event\_date, event\_type) where event\_type may assume values according to the following list:
  - S1= birth
  - S2= diagnosis
  - S3= transfer in
  - S4= start (generic reason)
  - E1= death
  - E2= loss to follow up
  - E3 = transfer out
  - E4= end (generic reason)
  - P= generic episode
- add missing start events (birth, diagnosis) included in the merge table to the event table
- for each episode in the merge table, create a new event and add it to the event table with event\_type=P
- sort the event table by patient id, date, then parse it in order to detect and correct wrong patterns (see table below). At the end of the parsing process, each group of episodes will be included in a time interval delimited by a start and an end event.

Wrong pattern	Correct pattern	Corrective actions
P-S1	P	birth event after an episode are deleted
P-S2	P-E4-S2	a fake end event is included between an episode and a transfer event. The event date is the same of the episode
P-S3	P-E4-S3	a fake end event is included between an episode and a transfer event. The event date is the same as the episode date
P- <sub>-</sub>	P-E4	a fake end event is included after an episode if this is the last event for the patient. The event date is the same as the episode date
P-P	P-E4-S4-P	a couple of fake start and end event is included between to episode if the difference between thier dates is more than 365 days
E1-P	P	a death event before an episode is deleted
E1-Sa	E1	a start event after an episode is deleted
E2-P	E2-S4-P	a fake start event is included between a transfer event and an episode. The event date is the same of the episode
E3-P	E3-S4-P	a fake start event is included between a loss-to-follow-up event and an episode. The event date is the same of the episode
Ea-Ea	Ea	two consecutive end event (of any type) are replaced by only one end event (the second is discarded)
Ea-S1	Ea	a birth event following an end event (of any type) is deleted
Sa-Sa	Sa	two consecutive start event (of any type) are replaced by only one start event (the first one is discarded)
S- <sub>-</sub>	-	trailing start event are deleted
-E	-	leading end event are deleted
-P	S4-P	a fake start event is included before a leading episode. The event date is the same of the episode

- if an event is deleted while parsing the patient history, the event is logged into the activity\_quality table as not admissible.
- create the original activity table using the diary: remove episodes from the list of events and couple, for each patient, consecutive start and end events into fresh activity table records.

**Activity table quality check report**

The quality check report for the activity table includes the following items:

- Total number of :
  - missing values (absolute value and percentage)
  - values with wrong format (absolute value and percentage)
  - out of range values (absolute value and percentage)
  - incoherent values (absolute value and percentage)
- Distribution of:
  - missing values over each field (absolute value and percentage)
  - values with wrong format over each field
  - out of range values over each field
- Detailed list of errors:
  - values with wrong format
  - out of range values
  - incoherent values

**Population Table and Diabetic Population Table Quality Check**

Due to the simple structure of the population table and the diabetic population table, the quality check consists of a simple data format check (the same applied to the merge table and the activity table as a first step). Missing values, values with wrong format and out of range values are detected, then stored into tables named “population\_table\_quality” and “diabetic\_population\_table\_quality”.

Currently the system does not perform any coherence check among values in the same record and values among different records. Further improvements are possible to detect records with duplicate key (year, age banding).

**3.8 Statistical Engine Preprocessing**

Once the data has been imported into the BIRO Database and properly cleaned, the statistical engine can process the BIRO Database in order to produce the statistical report.

Statistical processing may use substantial system resources, particularly in terms of memory usage. That is due to the internal mechanisms of the R language, which loads the whole dataset in memory before performing any statistical processing. This way memory usage may grow very quickly for large datasets.

In order to speed up the statistical engine process, at the same time reducing the memory load and improving the scalability of the system with large datasets, some processing overhead has been pulled out from the statistical engine and is now handled directly by the database engine.

The database engine triggers low level PostgreSQL queries to produce essential subsets of the original data that can be more easily handled by the statistical engine. Based on the user selection for the year of interest, time interval and reference date, the database engine selects the cohorts of active patients and extracts the target subset of episodes at the outset.

Two different selection procedures have been developed for the scope:

- when a reliable activity table is available, the cohort of active patient is determined by selecting all patients with a record in the activity table corresponding to a start date anterior

to the end of the year of interest AND an end date posterior to the first day of the year of interest

- otherwise the cohorts of active patients can be selected by extracting those patients who present at least an episode within the year of interest

### 3.9 Embedded central engine

The current version of the BIRO system has been provided with some of the central BIRO system functionalities. The central engine has been included in the system in order to allow the user to print a global report from data included in multiple data sources made locally available. Assuming that multiple set of statistical objects referred to different sub data sources are available on the local machine, the BIRO system can process the entire group to produce a .pdf and .html global report, as described in Figure 11.

As for the Statistical Engine, also the Central Engine requires a structured database at input. The database is created and populated by a small tool named BIRO “CSVImporter”. The tool searches for the statistical objects within a specified source folder and loads all objects into the BIRO Central Database.

It is important to remark that the BIRO Central Database is completely different from the BIRO Output Database used by the Statistical Engine.

Further specifications are needed to make possible for the central engine to operate not only at the global level, but also locally. A precise rule to create a registry of local data sources, so that the central server can always understand the true origin of the data, is needed to be designed. This activity would allow to finalize a product that would be able to run in an identical way both locally and centrally. This way, the BIRO system will present a complete recursive structure, through which the central engine can process aggregate data repeatedly to carry out European reports. Final update of the BIRO system will include such functionalities, together with a server side communication software that will listen a record all transfers with a precise set of rules across the network.

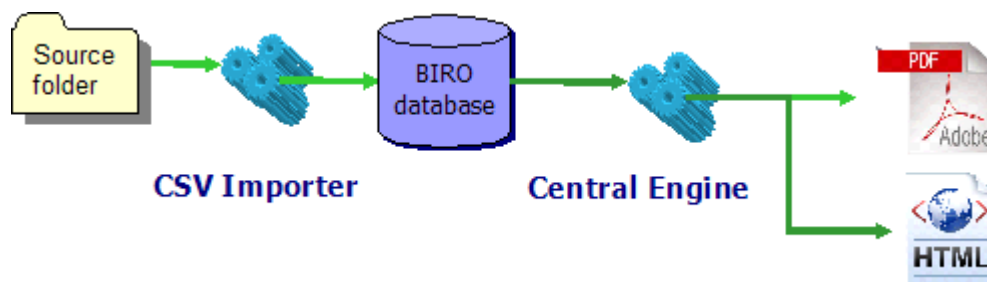


Figure 11: Central engine

## 4. Results

The final product of the BIRO system release 2010 includes architectural changes as well as a renewed graphical user interface. Some modifications turned out to be indispensable in order to mirror changes applied to the Database Engine. Other updates were based on users feedback collected during the test and training sessions, which were quite useful to improve the usability of the system. The requests of many users focused on the need to inspect local data, in particular the result of the import process, as well as outputs delivered by the statistical engine. The present chapter aims to describe the BIRO System from a user perspective.

### 4.1 Working directory

At startup, the BIROBox asks the user to select a folder as “Working Directory”.

The working directory should be a specifically dedicated folder where the system:

- stores all the outputs (exported XML archives, log files, statistical objects, statistical reports)
- searches for all input files (CSV, XML, customized transformations)
- saves all configuration files



**Figure 12: BIROBox startup panel**

The working directory is organized into a hierarchy of sub folders: a configuration folder plus a folder for each specific work package (database engine, statistical engine, communication software, etc.). Since everything referred to the BIROBox is contained in the working directory, the user may easily backup or share a particular configuration. That means that the user can create multiple working directories. When the user switches from a working directory to another, the BIROBox assumes that settings have been saved into it, otherwise it creates a new fresh configuration.

Usage of BIRO within the BIROX distribution made clear that the working directory is an indispensable feature. The definition of a working directory allows the system to keep local files and settings away from the source code of the software, so that they can be easily updated if necessary without losing the work done.



## 4.2 BIROBox layout

Similarly to the previous version, the main window of the BIROBox displays a button panel on the left, allowing to access all the main functions of the System:

- Setup (new): this section contains some global settings for the BIROBox
- Database Engine: this section includes all data management functions: import, quality check, export and inspection. Joins the old section of the “BIROAdaptor” with that of the “BIRODatabaseManager”
- Statistical Engine: in this section the user may configure and trigger the statistical calculation as for the previous version. Here the user may also browse the results of statistical processing.
- Central Engine (new): this section allows the user to run statistical analysis on multiple sets of statistical objects resulting from the analysis of different datasets
- Communication Software: as for the previous release, this section allows the user to select Statistical Objects and send them to the Central Server for the compilation of the global report

### 4.3 Setup panel

The setup panel shows the current working directory. It allows the user to set the credentials to access the underlying PostgreSQL BIRO Database and those of the external input database to be used as data source.

Since the number of input datasets has increased since the previous BIROBox release, it was necessary to extract the database credential settings as a common rule for all the datasets configuration. No changes have been done regarding the selection of the DBMS driver or the creation of custom driver.

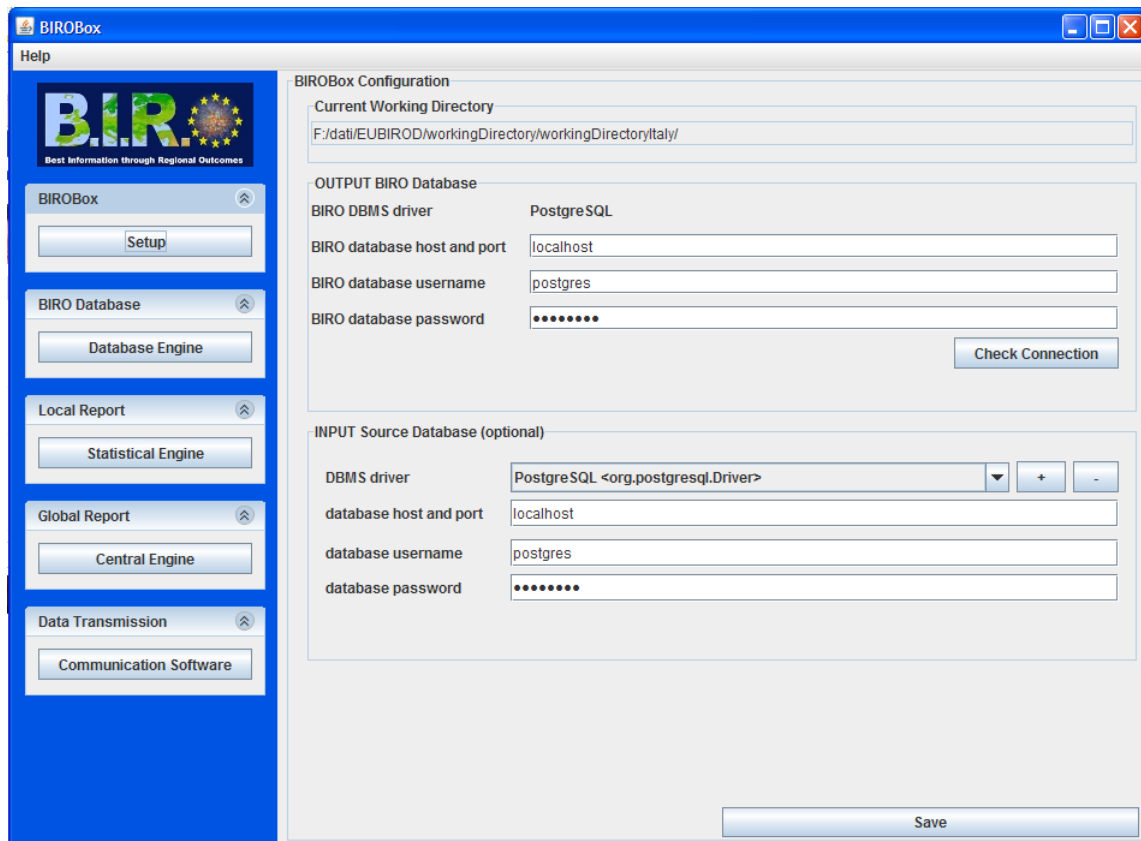


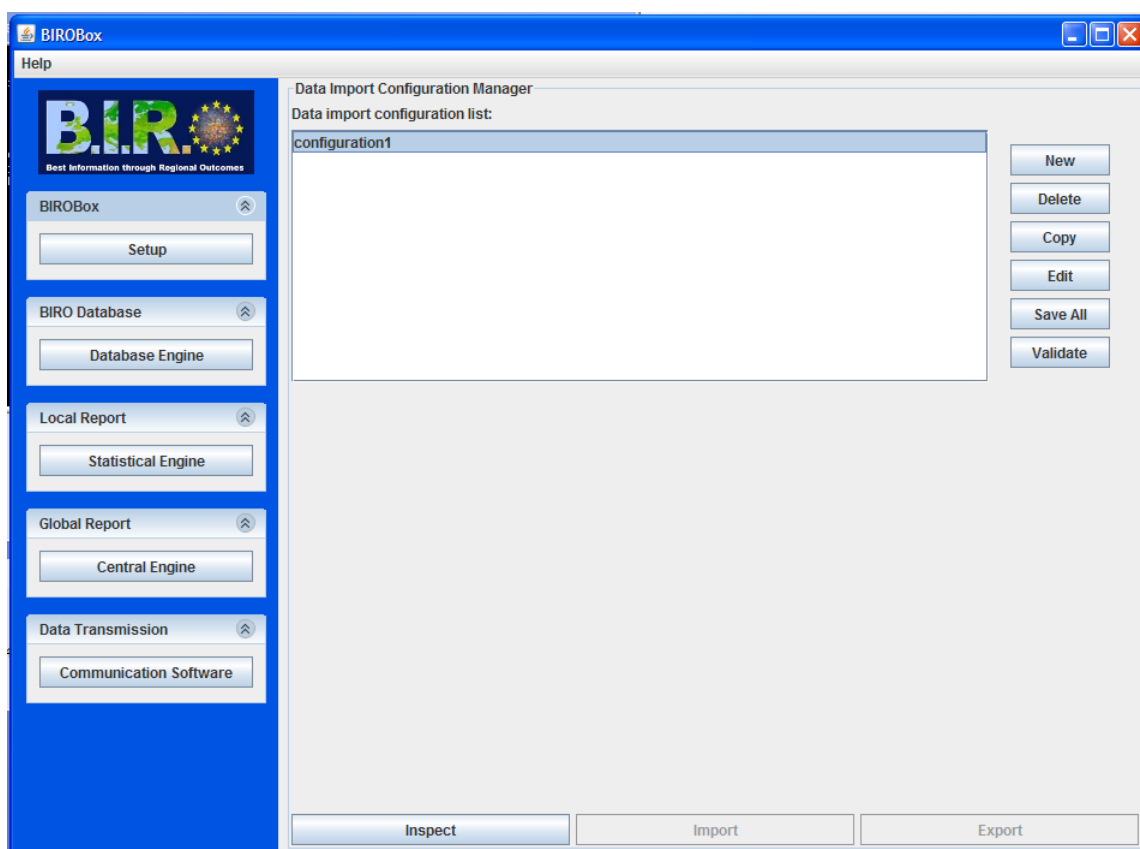
Figure 13: BIROBox setup panel

## 4.4 Configuration Validator

A new validation feature has been added to the section of the Configuration Manager Panel of the Database Engine. The import/export function is enabled only if the selected configuration successfully passes the validation test, triggered by clicking the “Validate” button on the right button panel.

Every time the user edits a configuration, it must be validated again before running the import/export.

Validation allows detecting potentially incorrect configurations before the import/export process starts. Since these processes can take a long time to be completed, the user must be notified to make an optimal usage of the available time.



**Figure 14: Configuration manager**

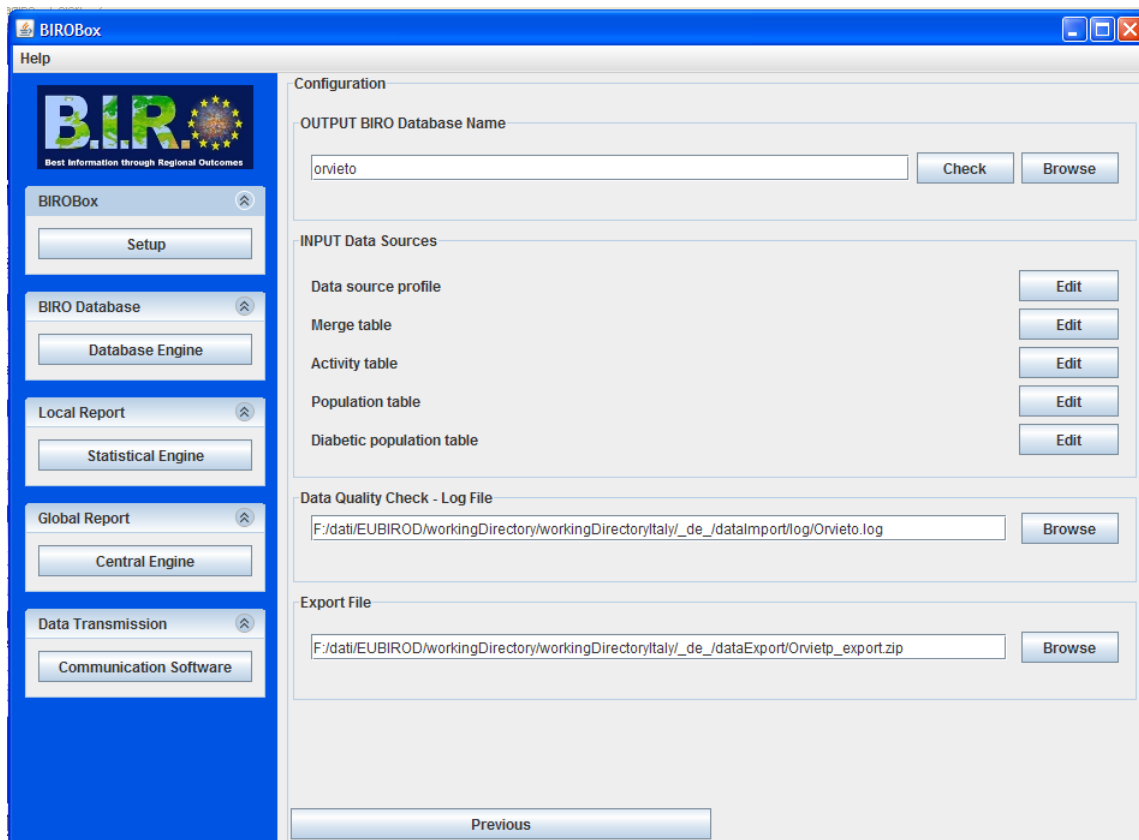
The configuration validator verifies the following:

- all the selected file paths exist (for CSV files, log files, import and export XML archives, transformation files)
- BIROBox can connect successfully to the BIRO Database (credentials are correct and the database exists)
- BIROBox can connect successfully to the Input Database, if any (credentials are correct, the database exists, the selected table exists)
- all the BIRO fields marked for extraction are mapped to an existing local field

- all the mandatory fields in the Data Source Profile have been set

Although the validator can trap the most common configuration issues, the validation cannot be exhaustive. The validator cannot perform any test on the correctness of the mapping. Choosing the right pattern for date fields, the unit of measurement for numeric field, the enumerated values for enumerated fields are all left to the end user. The validation does not free the user from the obligation of a careful mapping for each field.

## 4.5 Configuration Editor



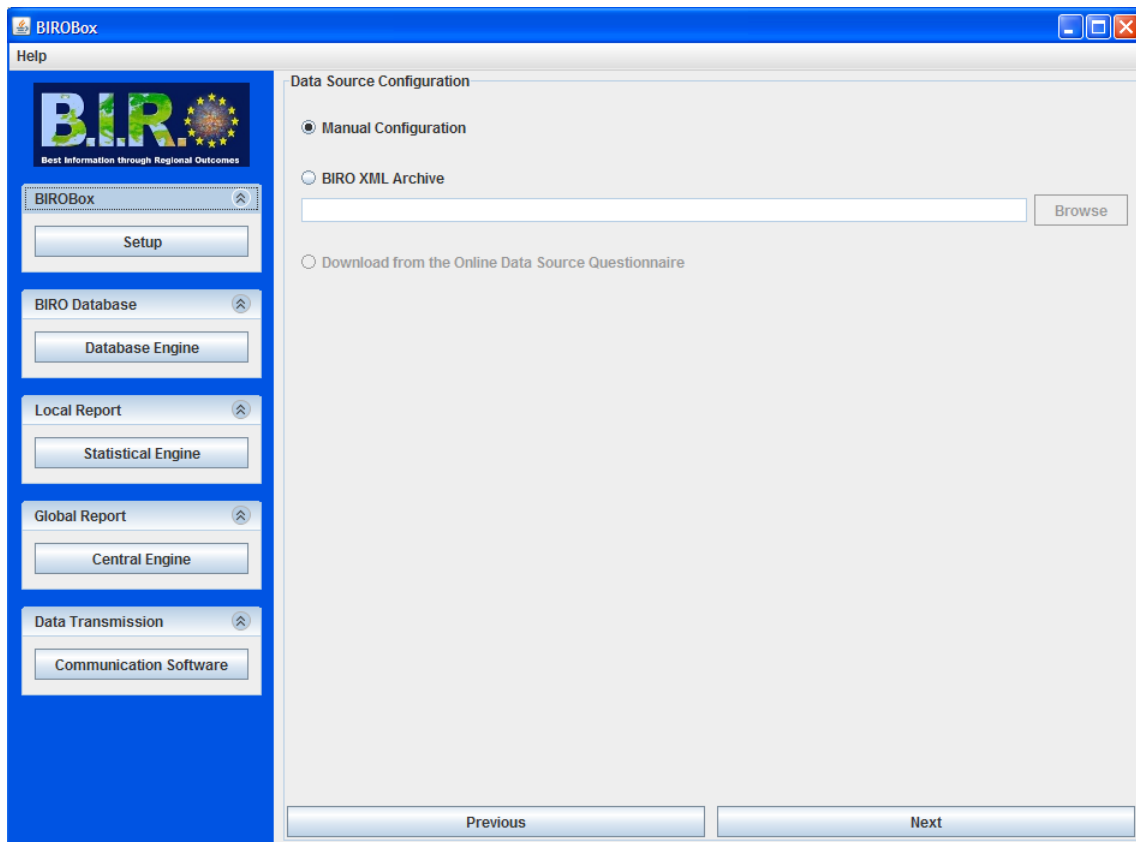
**Figure 15: Configuration editor**

Each time the user selects the “Edit” button in the Configuration Manager Panel, the BIROBox displays a completely revised wizard (see Figure 15).

Several important changes have been made in the current version:

- the user may browse the underlying PostgreSQL platform and select an existent database to be used as output BIRO Database; or, he/she may easily create a new database without interacting with the PostgreSQL interface
- this panel is the starting point for the configuration of all data sources (data source profile, merge table, activity table, population table, diabetic population table). Each sub section can be accessed through the corresponding “Edit” button on the left. Only the configuration of the data source profile, the merge table and the population table are mandatory.
- the user may select the path for the data quality log file, i.e. the report produced at the end of the data quality check, including basic quality statistics
- the user may select the path for the XML export archive

### 4.5.1 Data source profile configuration



**Figure 16: source profile input configuration**

The panel allows choosing if the BIROBox receive information on the data source manually or through a BIRO XML Archive.

The user may decide to import the site information directly from a previously exported BIRO XML archive, if available, by selecting the appropriate option and browsing for the correct file path. Also the XML summary file obtained at the end of the Online Data Source Questionnaire can be profitably used as a site profile source within the BIROBox.

Otherwise, the user may leave the default option and proceed with the manual configuration, clicking the “Next” button in the navigation bar.

## 4.5.2 Data input configuration

BIROBox

Help

Best Information through Regional Outcomes

BIROBox

Setup

BIRO Database

Database Engine

Local Report

Statistical Engine

Global Report

Central Engine

Data Transmission

Communication Software

Merge Table Source Configuration

☐ database

database name

database table

Browse

☒ csv file

csv file name \*

F:/dati/EUBIROD/workingDirectory/workingDirectory/Italy/\_de\_/dataimport/csv/umbr

Browse

separator

|

☐ xml file

archive file name

Browse

Customized Toolbox

☐ Customized configuration available

Browse

Previous

Next

**Figure 17: Data input configuration**

When editing the configuration of the input data sources, the user may choose among three different source types: a database table, a CSV file, or an XML file (BIRO format).

If a transformation file for the customized toolbox is available, then the user may enable the corresponding option and browse the file.

Since the Customized Toolbox can be only applied to CSV files, this option is disabled if the user switches to a database or an XML source.

### 4.5.3. Field Mapping Configuration

BIROBox

Help

Best Information through Regional Outcomes

BIROBox

Setup

BIRO Database

Database Engine

Local Report

Statistical Engine

Global Report

Central Engine

Data Transmission

Communication Software

Fields mapping configuration

Configure mapping between BIRO fields and local fields

BIRO field

BIRO field name: BIGUANIDES

BIRO field code: BIRO056

BIRO field description:

Patient recorded as receiving Biguanide treatment

☒ Extract from local database

Local field name

Metformina

BIRO category	Expression	Local value	BIRO Value
No Biguanide therapy	is custom text	0	0
Biguanide therapy	is custom text	1	1

Previous

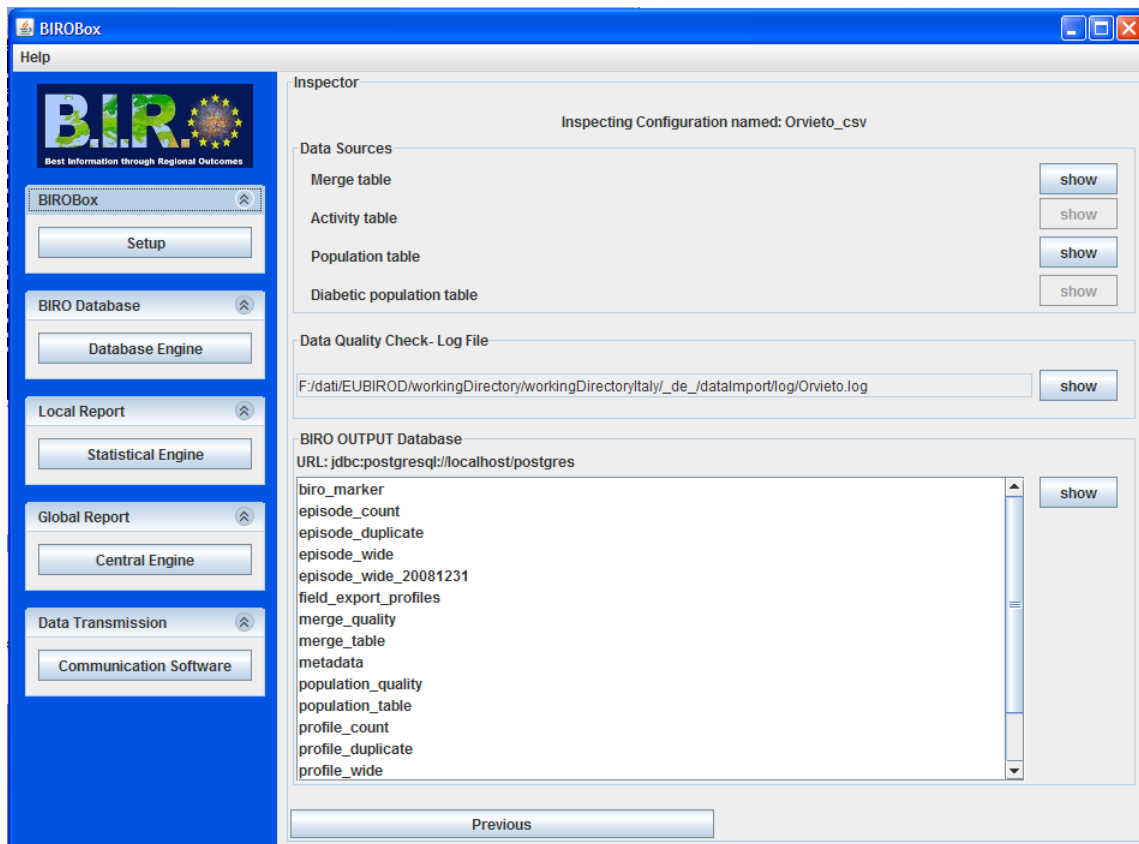
Finish

Figure 18: Field mapping configuration

If the user chooses a database table or CSV file as data input, but no customized transformation is available, then he/she shall map BIRO fields manually as in the previous release.

Slight improvements were made to the panel to facilitate the mapping, e.g. text descriptions from the BIRO Data Dictionary and a field inspector allowing the user to display the content of the selected local field (maximum 50 rows from the database).

## 4.6 Inspector



**Figure 19: Inspector**

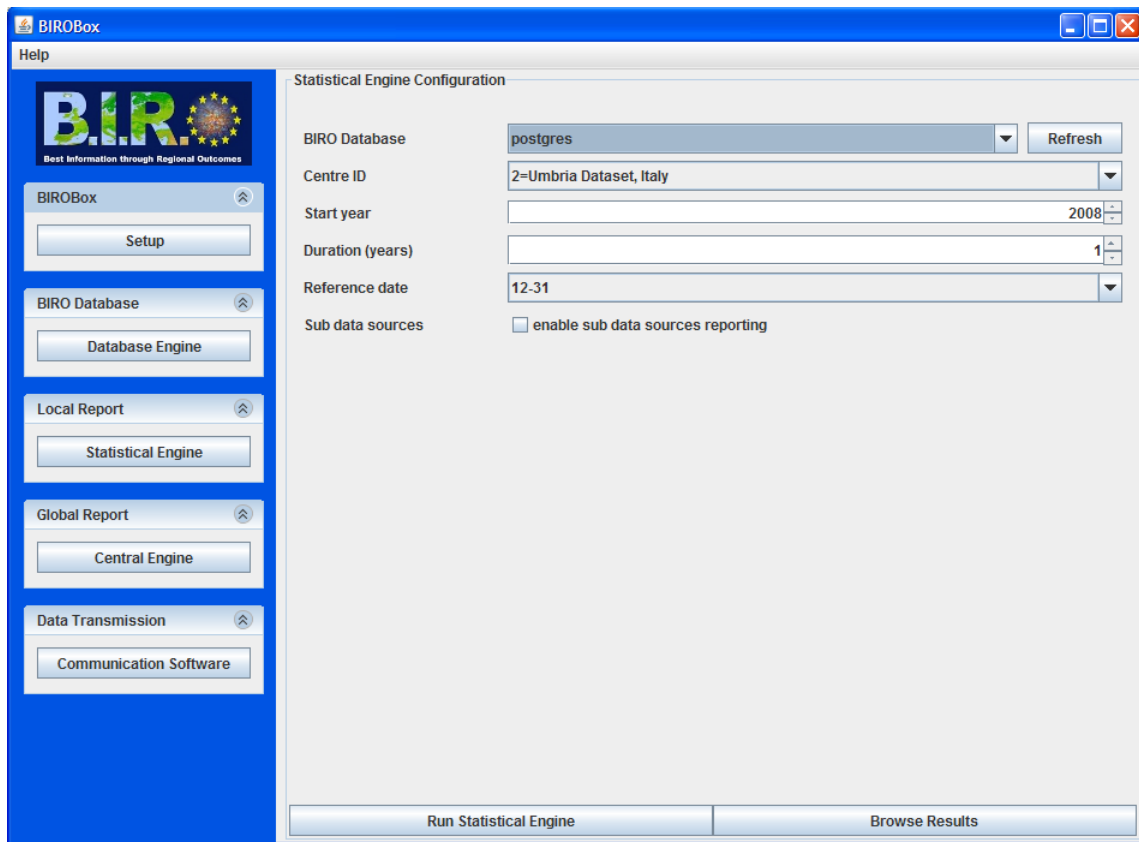
The inspector is a new sub section of the Database Engine allowing the user:

- to visualize the content of the input data sources
- to explore the BIRO database with no need of pgAdmin, the PostgreSQL Admin interface
- to inspect the results of the mapping and the import processes
- to display the data quality log file

The inspector can allow the user to browse inputs and outputs of the Database Engine in the same panel, allowing an immediate comparison of all files.



## 4.7 Statistical engine panel



**Figure 20: Statistical engine**

The sub data source reporting option has been added to the statistical engine panel configuration.

The option triggers an important feature of the BIRO system: when data from different centres are available at a single data source, the user may decide to force the statistical engine to compute all indicators for each sub data source separately.

In this case, the statistical engine groups episode data, based on the value of the variable used as a “sub data source”, to produce stratified indicators for each level of that variable.

## 4.8 Statistical engine browser

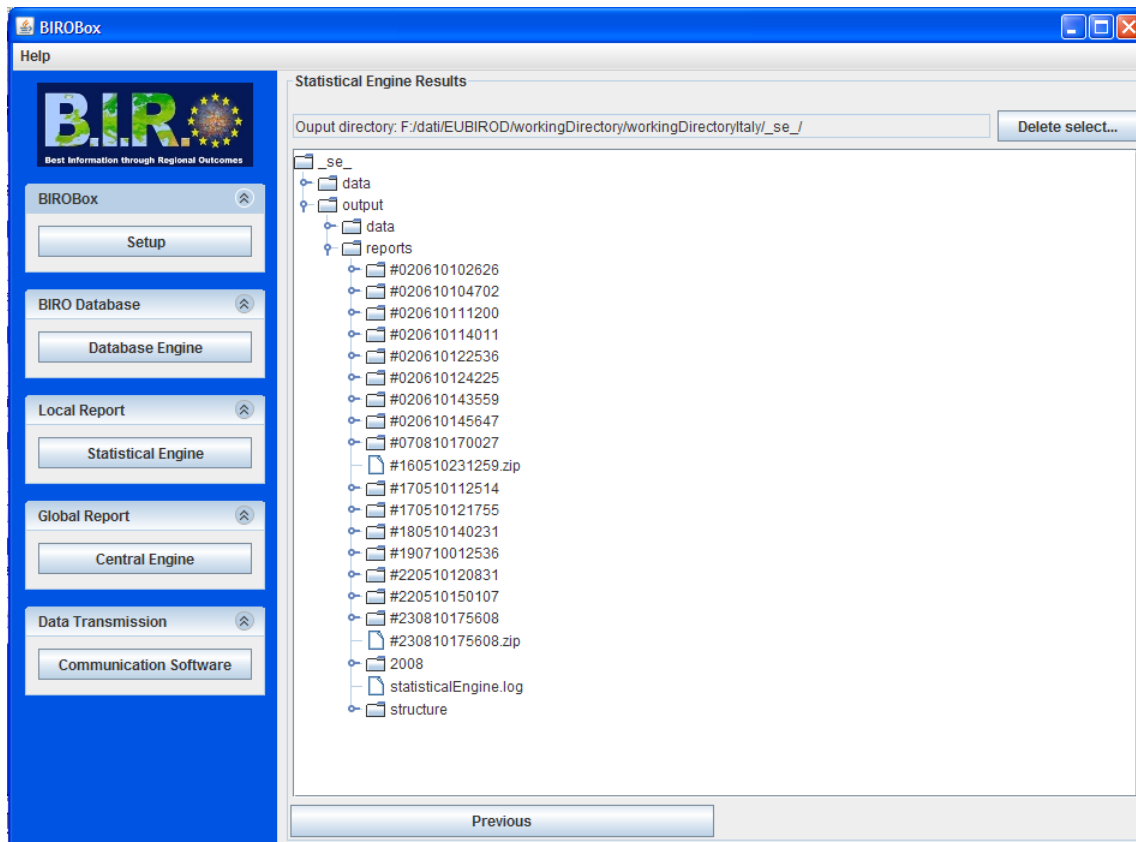
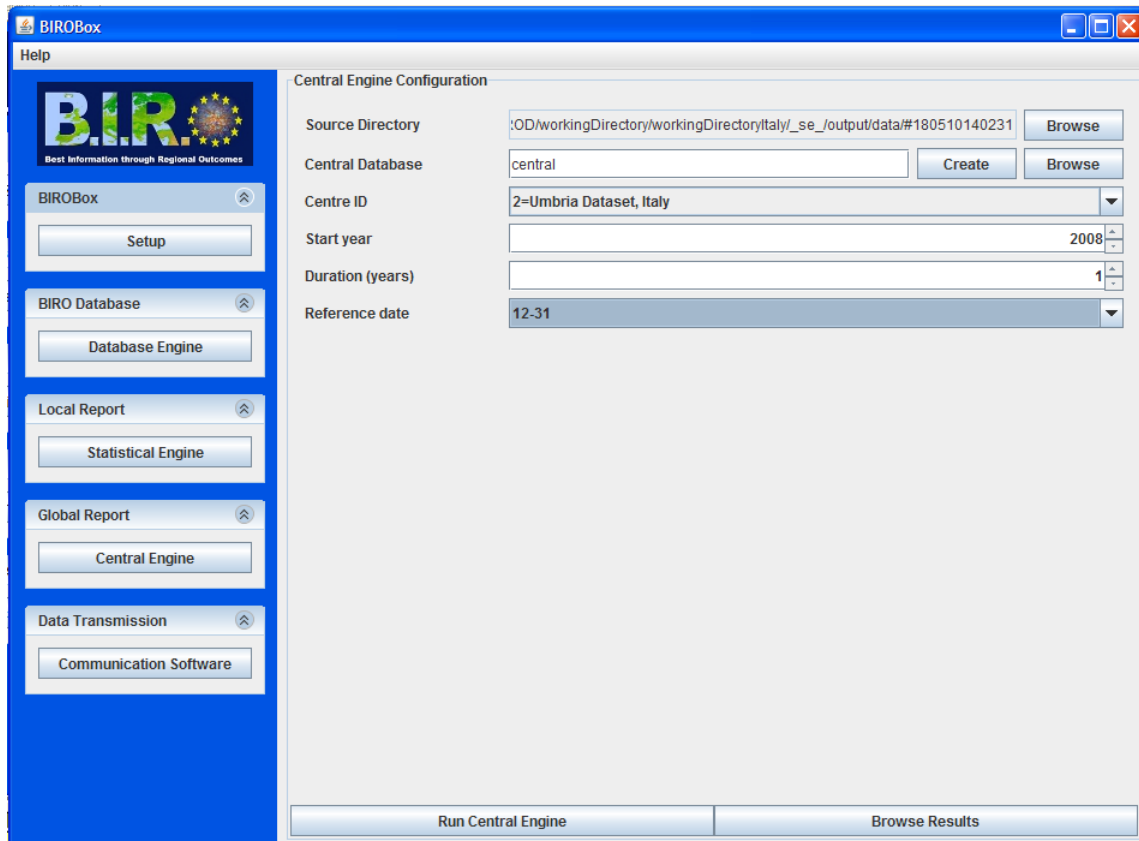


Figure 21: Statistical engine browser

By selecting the “Browse” button in the Statistical Engine panel, the BIROBox displays the Statistical Engine browser, i.e. a window listing all the statistical reports ever produced. Every time the user run the analysis, the statistical engine creates a new folder named with a corresponding “timestamp”. Within each folder the user will find the following items:

- ➔ *data*: folder containing CSV export of patient profiles and episodes representing the statistical engine cohort.
- ➔ *output*
  - ➔ *data*: folder containing the CSV files representing the statistical objects (aggregated data to be sent to the central engine)
  - ➔ *report*: folder containing the statistical reports and all its components
    - ➔ *#timestamp*
      - ➔ *year*
        - ➔ *data source id*
          - ➔ *report.html*: the index page of the HTML report in HTML format
          - ➔ *report.pdf*: the report in PDF format
          - ➔ *graphs*: folder containing all the charts in the report in SVG, PDF, JPG, PNG format
          - ➔ *html*: folder containing the HTML pages of the report
          - ➔ *pdf*: folder containing all the latex component of the PDF report
          - ➔ *tables*: folder containing all the tables in the report separately in HTML format
          - ➔ *wp*: folder containing all the HTML pages of the report formatted for the Web Portal

## 4.9 Central engine panel



**Figure 22: Central engine**

The Central Engine panel allows configuring and running a global statistical analysis on multiple statistical objects and browsing the results obtained.

The layout is exactly the same as the Statistical Engine Panel. The configuration panel asks the user to specify the year(s) of interest for analysis and the database to be used as a basis for calculations. Moreover, the system requires the user to specify the source directory where the statistical objects are stored.

By selecting the “Run Central Engine” button, the user will trigger the import of statistical objects into the selected database and then the statistical process. The results will be stored into a folder named “\_ce\_” within the Working Directory.

The central engine results browser can be accessed by selecting the “browse button” exactly as for the Statistical Engine.

## 5. Discussion

### 5.1 Consortium feedback

The first release of the new version of the BIRO system was delivered to the EUBIROD Consortium in April 2010. All partners were asked to install the software, feed the BIROBox with local data, analyse the statistical reports and send feedback to the Coordinating Centre.

Results were presented at the Special BIRO Academy meeting, held on the 4<sup>th</sup>-5<sup>th</sup> June 2010 in Rome 2010. A total of N=14 partners presented the results of their local analysis, together with an evaluation of their experience with the BIRO software and the difficulties that they have encountered.

All presentations are publicly available on the BIRO Academy web site, at the following address: [http://www.eubirod.eu/academy/special\\_meeting/special\\_meeting\\_lectures.html](http://www.eubirod.eu/academy/special_meeting/special_meeting_lectures.html)

An initial analysis of the central engine was presented, for an overall sample of approximately 50,000 diabetic subjects.

Almost all partners successfully delivering the analysis noticed a substantial improvement in the user-friendliness of the system and appreciated the way the BIROBox had been reshaped.

Based on the discussion held at the meeting and the general feedback received, the following suggestions were endorsed for further development:

- to make available more documentation on the BIRO System, especially a detailed user guide, helping the user to understand how to configure the system and how to interpret the statistical results. Since the quality and the completeness of the report strongly depends on the level of compliance of the local data with the BIRO dataset, many partners asked for a support documentation on how to prepare data for the BIROBox
- to add a section on the paediatric population, still not well described. The majority of the BIRO fields measure diabetes outcomes or risk factors (e.g. smoking, alcohol intake) that are not applicable to the paediatric population. Other fields can be useful to better analyse type 1 diabetic patients (e.g. Coeliac Disease, Hashimoto Thyroiditis, autoimmune diseases,...). Consequently, lower and upper boundaries for the BIRO fields must be revised accordingly: although valid for adults, current thresholds do not apply to paediatric patients and may lead to discarded values. The same applies to age bandings, as the paediatric population is almost completely contained in the first age band (0-14). Splitting the first range would allow a fine grained description and therefore a better comprehension of the type 1 diabetes. Additional age banding would be: 0 – 4; 5 – 9; 10 – 14.
- to add the possibility to combine data coming from multiple data sources by appending them to the same BIRO Output Database. Currently the multiple data sources may be only analysed separately using different BIROBox Configuration
- to broaden the coverage of the BIRO Common Dataset to include more information on social and emotional determinants
- to increase the capability of running complex analyses on large data sets more quickly and efficiently

## 5.2 Conclusions and perspectives

Feedback received at the Special BIRO Academy meeting confirm that the current release of the Database Engine and the associated BIROBox has definitely improved the possibility for EUBIROD partners to manage local data and obtain statistical results. Currently, the majority of partners are able to deliver complete reports on their own.

Further improvements have been made to make the process easier and more sophisticated, particularly with regards to extensive quality checks.

The final part of the work undertaken can now focus on the usage of the BIRO system for the delivery of European Diabetes Report. These operations will allow to further refine and test the usage of the system.

Improvements on the agenda include:

- better management of the paediatric population and development of an ad-hoc paediatric common dataset. A revised version of the Common Dataset (Deliverable D5.2) already includes new candidate fields that now await confirmation from EUBIROD clinical experts
- the system can be parametrized to generalize the approach to other chronic diseases. This option would require to eliminate all hard coding from the source code and use external plugins to populate specific areas of the BIROBox, particularly mapping. The statistical engine should be profoundly revised accordingly.
- agile reporting: new options may allow to select specific subset of indicators or subjects corresponding to target strata. This way also the report would be more focused and small in size.
- smarter dataset mapping functions: ETL features may be improved and further tested in real life conditions where the potential for data linkage are extremely varied and the average register administrator has limited skills and a lack of resources to adapt new software
- delivery of the server side: the BIROBox can be conveniently used also from a server side by each national coordinator, so that the central workload would decrease and the local awareness/participation would be enhanced. To allow this, a strict coding structure and a hierarchical registry of European data sources must be established and embedded in the BIRO system

By using the BIROX distribution, package, the BIRO system can be easily and quickly updated, so that any future improvement can be timely made available to all partners with minimal effort.

While the new release is currently being deployed, part of the improvements described above will be included in the agenda for the last year of the project.

## References

1. The EUBIROD Project, deliverable D5.2 Privacy Impact Assessment, available at:  
[http://www.eubirod.eu/documents/downloads/D5\\_2\\_Privacy\\_Impact\\_Assessment.pdf](http://www.eubirod.eu/documents/downloads/D5_2_Privacy_Impact_Assessment.pdf)
2. The EUBIROD Project, deliverable D5.1 EUBIROD Common Dataset, available at:  
[http://www.eubirod.eu/documents/downloads/D5\\_1\\_Common\\_Dataset.pdf](http://www.eubirod.eu/documents/downloads/D5_1_Common_Dataset.pdf)
3. The EUBIROD Project, deliverable D4.1 Report on Training, available at:  
[http://www.eubirod.eu/documents/downloads/D4\\_1\\_Report\\_On\\_Training.pdf](http://www.eubirod.eu/documents/downloads/D4_1_Report_On_Training.pdf)
4. The BIRO Project, deliverable D6.1 Database Engine, available at:  
[http://www.biro-project.eu/documents/downloads/D6\\_1%20Database%20Engine.pdf](http://www.biro-project.eu/documents/downloads/D6_1%20Database%20Engine.pdf)
5. The BIRO Project, deliverable D6.2 Database Engine Update, available at:  
[http://www.biro-project.eu/documents/downloads/D6\\_2\\_Database\\_Engine\\_Update.zip](http://www.biro-project.eu/documents/downloads/D6_2_Database_Engine_Update.zip)
6. The EUBIROD Project, deliverable D7.1 Customized Toolbox, available at:  
[http://www.eubirod.eu/documents/downloads/D7\\_1\\_Customised\\_Toolbox.pdf](http://www.eubirod.eu/documents/downloads/D7_1_Customised_Toolbox.pdf)
7. Pentaho Data Integration, available at: <http://kettle.pentaho.com/>