PROJECT N. 037033

EXIOPOL

A NEW ENVIRONMENTAL ACCOUNTING FRAMEWORK USING EXTERNALITY DATA AND INPUT-OUTPUT TOOLS FOR POLICY ANALYSIS

THE EXIOPOL DATABASE MANAGEMENT SYSTEM

–

COMPUTATIONAL STRUCTURE

Report of the EXIOPOL project

Title	The EXIOPOL database management system – Computational structure
Purpose	Description of the computational structure, with all steps and intermediate results described
Filename	EXIOPOL_DIII 4 b-2_FINAL.doc
Authors	Arjan de Koning, Reinout Heijungs, Gjalt Huppes
Document history	not relevant
Current version.	version 2
Changes to previous version.	not relevant
Date	15 October 2008
Status	Draft
Target readership	Cluster III partners
General readership	not relevant
Dissemination level	Cluster III partners

Arjan de Koning, Reinout Heijungs, Gjalt Huppes
Institute of Environmental Sciences (CML) - Universiteit Leiden

October 2008

Prepared under contract from the European Commission

Contract no 037033-2
Integrated Project in
PRIORITY 6.3 Global Change and Ecosystems
in the 6th EU framework programme

Deliverable title:	The EXIOPOL database management system – Computational structure
Deliverable no. :	D III.4.b-2
Due date of deliverable:	Month 18
Period covered:	from 1st March 2007 to 1st September 2008
Actual submission date:	10.10.2008
Start of the project:	01.03.2007
Duration:	4 years
Start date of project:	01.03.2007
Project coordinator:	Anil Markandya
Project coordinator organisation:	FEEM

Executive Summary

This report gives an overview of the different computational steps that are related to the EXIOPOL database system. There are six such steps:

- an import script;
- an import consistency check;
- a data consistency check;
- a RAS script;
- an IO script;
- an export script.

These steps are described with respect to their input, function, and output. Together, they link the input data as are composed from different sources externally into the three main databases of the EXIOPOL database system, and link the database system to several of its applications.

Table of contents

1 Introduction

1.1 Overview of WS III.4

This report is one of a series of reports that describe the EXIOPOL database management system that is developed in Workstream III.4 of the EXIOPOL project. The full series consists of the following reports and other deliverables.

Number	*Title or description*
DIII.4.a-2	**Trade Linking Method** Method for trade linking described
DIII.4.a-3	**Four Country Test** Algorithm available and applied on a test of 4 countries
DIII.4.b-1a	**Data Input Protocol** The protocols for data input are described after development in discussion with the leaders of WP.III.2.a/b and WP.III.3.a/b. The data input protocol shall link easily with the form in which data are made available after the transformations in Block 1 (i.e usually matrices in Excel, or Access)
DIII.4.b-1b	**EXIOBASE Main Design** Provision of a report that provides an extensive list of design criteria of the database, 4-5 strategic design options, and proposes a design option. This report shall reflect the holistic perspective on the database development work as discussed under 'Objectives'
DIII.4.b-1	**Nomenclature and Transformations** Description of the basic nomenclature used for all output data, with transformation tables to classifications as used in all data sources
DIII.4.b-2	**Computational Structure** Description of all steps and intermediate results
DIII.4.b-3	**EXIOBASE Software** Full Database Manager
DIII.4.b-4	**EXIOBASE Data** Database filled with all data
PDIII.4.c-1	**Set-Up of World Trade Model Interface** Preliminary report on interface to World Trade Model and its extensions
PDIII.4.c-2	Outline on interface to economy-wide and bottom-up sectoral models
DIII.4.c-1	**World Trade Model Interface** Interface and report on World Trade Model and its extensions
DIII.4.c-2	Interface and report on economy-wide and bottom-up sectoral models

The present report is DIII.4.b-2. It is devoted to a description of the computational steps that are related to the EXIOPOL database system.

1.2 Overview of the database system

From DIII.4.b-1b (De Koning et al., 2008a), we reproduce **Figure 1.1** below, which gives an overview of the main architecture of the EXIOPOL database system.

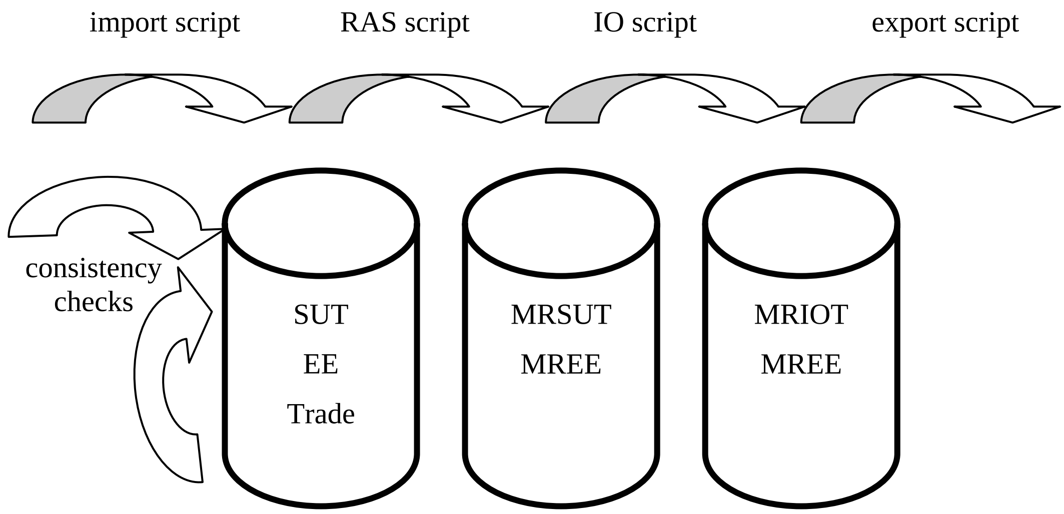


Figure 1.1: Database (or multiple database) structure of the EXIOPOL database. Legend: cylinders represent databases; grey arrows represent data flows; white arrows represent consistency checking procedures.

The three cylinders represent the databases of the EXIOPOL database system. These have a certain internal architecture, for instance, they consist of a number of linked tables, following an entity-relationship model (see DIII.4.b-1b; De Koning et al., 2008a). The flow of data into and out of each database, and especially the steps that carry information from one database into a next database requires computational steps. Some of these are straightforward, but others are more complicated. Moreover, in most cases these computational steps can be carried out in a number of different ways, according to the modeling assumptions made, or depending on the type of application intended. The following section gives an overview of these computational steps.

1.3 Overview of the computational steps

In **Figure 1.1**, we can discern six computational steps. These are:

- the import script: tables with records of data of the country level SUTs and other information are imported into the database system;
- the import consistency check: the data records that are imported are checked with respect to their conformity with a number of defined rules;
- the data consistency check: the collection of data records that reside in the first database are checked with respect to their conformity with a number of defined rules;
- the RAS script: the trade linking of individual country SUTs into a multi-region SUT, MRSUT;
- the IO script: the MRSUT is converted into a number of variants of MRIOT;
- the export script: data of the MRIOT is exported in a number of different formats for modelling applications.

These six steps are described in the six next chapters.

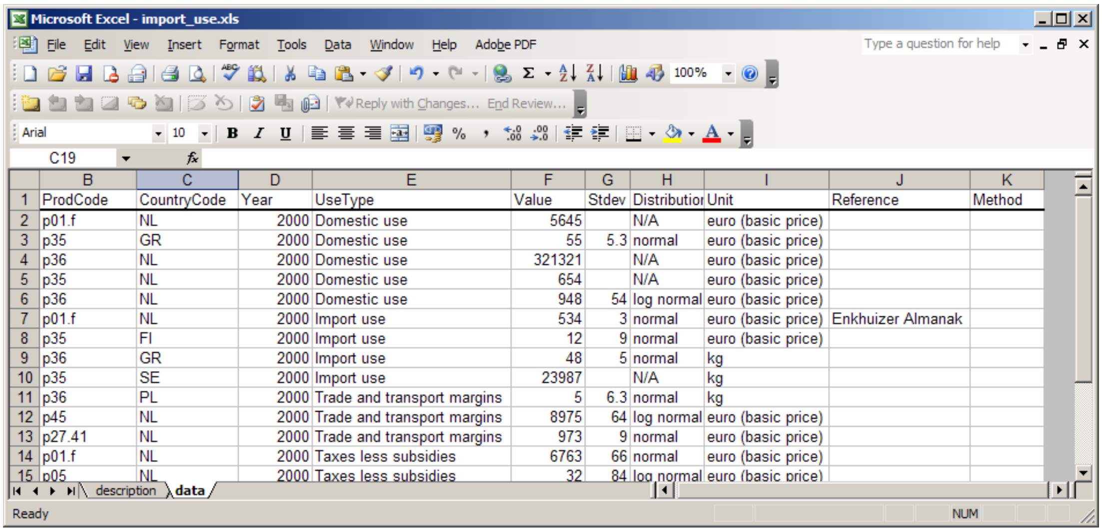
2 The import script

2.1 Purpose

The import script enables the import of tables with records of SUT-data and other information into the database system.

2.2 Input of the script

For the input of the import script, a template has been made available in Microsoft Excel. From DIII.4.b-1a (De Koning et al., 2008b), we reproduce a screenshot of the format of these data (Figure 2.1). This example is for the use table, but similar templates are available (and described in DIII.4.b-1a) for other data types.



	ProdCode	CountryCode	Year	UseType	Value	Stdev	Distribution	Unit	Reference	Method
2	p01.f	NL	2000	Domestic use	5645		N/A	euro (basic price)		
3	p35	GR	2000	Domestic use	55	5.3	normal	euro (basic price)		
4	p36	NL	2000	Domestic use	321321		N/A	euro (basic price)		
5	p35	NL	2000	Domestic use	654		N/A	euro (basic price)		
6	p36	NL	2000	Domestic use	948	54	log normal	euro (basic price)		
7	p01.f	NL	2000	Import use	534	3	normal	euro (basic price)	Enkhuizer Almanak	
8	p35	FI	2000	Import use	12	9	normal	euro (basic price)		
9	p36	GR	2000	Import use	48	5	normal	kg		
10	p35	SE	2000	Import use	23987		N/A	kg		
11	p36	PL	2000	Trade and transport margins	5	6.3	normal	kg		
12	p45	NL	2000	Trade and transport margins	8975	64	log normal	euro (basic price)		
13	p27.41	NL	2000	Trade and transport margins	973	9	normal	euro (basic price)		
14	p01.f	NL	2000	Taxes less subsidies	6763	66	normal	euro (basic price)		
15	p05	NL	2000	Taxes less subsidies	32	84	log normal	euro (basic price)		

Figure 2.1: Screenshot of the result of imported data for the entity quantities of intermediate use.

The SUT data is not imported as a matrix, but as separate records.

2.3 Function of the script

The script does not perform calculations. It gathers records, disassembles the information, and stores the data elements in the appropriate tables of the database system.

2.4 Output of the script

The script has no visible output. It adds records to a number of tables of the first relational database, so with appropriate software to view these tables (such as Microsoft Access), one may observe its output.

3 The import consistency check

3.1 Purpose

The import consistency script checks if the data records that are imported are in conformity with a number of defined rules.

3.2 Input of the script

The input of the import consistency check is the same as that of the import script itself. In fact, the two scripts are integrated: during import of data, the consistency of these data is checked. This import format is shown in Figure 2.1.

3.3 Function of the script

The script checks for a number of issues. These are:

- Is the file structure (number of columns, etc.) OK?
- Is the content of the columns with categorical data conforming to the nomenclature? In other words, are the names of products, industries, emissions, etc. conforming to the standard?
- Is the content of the columns with numerical data conforming to the conventions of numbers?
- Are the numbers within a valid range? E.g., are all use and supply data non-negative?

The script does not transform data.

3.4 Output of the script

The script performs checks, and has as its primary function to provide warnings. Invalidly formatted data files and invalid records should be flagged to the person who is responsible for the data import. Thus, the script should provide error messages that are understandable to that person, and that are helpful to detect and repair the origin of the error.

4 The data consistency check

4.1 Purpose

The data consistency script checks whether the collection of data records that reside in the first database are conforming to a number of defined rules.

4.2 Input of the script

The input of the data consistency check is the first relational database, containing the SUTs and additional information, such as environmental extensions.

4.3 Function of the script

The script checks for a number of issues. These are:

- Do the data agree with usual economic checks, such as the equality of row sum and column sum?
- Are there industries that use products but do not produce products, or the other way around?
- Are there industries that use more than they produce?
- Are there industries that have environmental extensions, but do not produce products, or the other way around?
- Are there products that are exported but not imported, or the other way around?

The script does not transform data.

4.4 Output of the script

The script performs checks, and has as its primary function to provide warnings. Invalid data sets should be flagged to the person who is responsible for the data import. Thus, the script should provide error messages that are understandable to that person. Detecting and repairing the origin of the error is more difficult than for the import consistency check, as it is always a combination of records that create the inconsistencies at this level. But at least, the error messages should indicate which industry/product/country is causing which problem.

5 The RAS script

5.1 Purpose

The RAS script performs the trade linking of individual country SUTs into a multi-region SUT.

5.2 Input of the script

The input of the RAS script is the first relational database, containing the SUTs and additional information, such as environmental extensions.

5.3 Function of the script

The script is an implementation of the generalized RAS (GRAS) algorithm described by Bouwmeester & Oosterhaven (2008). Basically, it combines data on domestic supply and use and data on import and export to transform a block diagonal SUT, where every block represents a domestic SUT, into a fully crowded multi-region SUT, where the use of foreign products by domestic industries, and the supply of domestic products to foreign industries is specified in a consistent way.

Moreover, the additional information that is not part of the RAS procedure (such as the environmental extensions) is carried over as well.

Various forms of the RAS script may be considered. For instance, there are several sets of assumptions that can be made, and the required level of convergence of the RAS procedure may be set as well. Currently, the procedures still requires some adjustment ‘by hand’. It will be fully formalized however. Which of the options will be set fixed, and which will be offered as a user-defined setting will become clear in the next few months of the EXIOPOL project.

5.4 Output of the script

The script has no visible output. It adds records to a number of tables of the second relational database, so with appropriate software to view these tables (such as Microsoft Access), one may observe its output.

6 The IO script

6.1 Purpose

The IO script converts the multi-region SUT into a multi-region IOT.

6.2 Input of the script

The input of the RAS script is the second relational database, containing the MRSUT and additional information, such as environmental extensions.

6.3 Function of the script

The script builds on the well-established theory of deriving IOT from SUT, which should not be very different from deriving MRIOT from MRSUT. The scope document (Tukker & Heijungs, 2008) describes the mathematical principles and some of the most important choices. These include:

- industry-by-industry format vs. commodity-by-commodity format;
- industry technology assumption vs. commodity technology assumption.

There are, however, more options. Ten Raa & Rueda-Cantuche (2003), for instance, describe more variants.

Besides the SUT to IOT transformation, additional information, such as the satellites, are to be cotransformed.

Which of these options will be set fixed, and which will be offered as a user-defined setting will become clear in the next few months of the EXIOPOL project.

6.4 Output of the script

The script has no visible output. It adds records to a number of tables of the third relational database, so with appropriate software to view these tables (such as Microsoft Access), one may observe its output.

7 The export script

7.1 Purpose

The export script takes care of the link between the database system and the “existing models”, such the World Trade Model. Data in the database can be exported in a number of different formats.

7.2 Input of the script

The input for the export script is at least the third relational database, containing the MRIOT and additional information, such as environmental extensions. There might be applications for which data of the second or even first database would need to be exported. To what extent this is really needed, is not yet clear at the present state of progress. Efforts will be directed to exporting the third database, and when an export of the other two databases might be necessary, we expect to be able to provide these with little additional effort.

7.3 Function of the script

The script should be very flexible in its output. To mention a few things:

- it should support the export of different types of data (IOT, factor inputs, environmental extensions, etc.)
- it should support different file formats (ascii, Excel, Matlab, XML, etc.);
- it should support different contents (IOT, coefficients table, Leontief inverse, etc.);
- it should support different file structures (e.g., industry name in first column, in first row, in last column, etc.);

Obviously, the number of options is large, too large in fact to serve within the project.

The use of SQL helps us in building a flexible export script. With SQL, we can build a number of export routines. These routines are provided as open source scripts, of which a local copy can be modified by a user for specific purposes.

Which export routines will be pre-defined in the EXIOPOL project will become clear in the next few months of the EXIOPOL project.

7.4 Output of the script

The script produces output files that contain the desired data in the specified format.

List of references

Bouwmeester & Oosterhaven, 2008) Technical report: The EXIOPOL database management system – Inventory of trade data and options for creating linkages. EXIOPOL DIII.1.a-3.

De Koning, A., R. Heijungs & G. Huppes (2008a) Technical report: The EXIOPOL database management system – main design. EXIOPOL DIII.4.b-1b.

De Koning, A., R. Heijungs & G. Huppes (2008b) Technical report: The EXIOPOL database management system – Protocols for data input. EXIOPOL DIII.4.b-1a.

Ten Raa, T. and J. Rueda-Cantuche (2003). The Construction of Input Output Coefficients Matrices in an Axiomatic Context: Some Further Consideration. Economic Systems Research 15, pp 439-455

Annex I: Contributors to the report

This report is the result of discussions between all partners in the EXIOPOL consortium. It has been edited and written by the following persons:

Arjan de Koning, Institute of Environmental Sciences (CML) - Universiteit Leiden

Reinout Heijungs, Institute of Environmental Sciences (CML) - Universiteit Leiden

Gjal Huppes, Institute of Environmental Sciences (CML) - Universiteit Leiden