Databox System

Central Data Management for Input-Output Models

User Manual and Technical Documentation

English

Wolfgang Boehme

wolfgang.boehme@economic-modeling.de

www.economic-modeling.de

4 November 2026

The documentation on this website was developed with the support of the AI Claude (Anthropic).

Table of Contents

1 Introduction **2**
- 1.1 What is the Databox? . . . 2
- 1.2 Concept . . . 2
- 1.3 Where Are the Data Stored? . . . 2
- 1.4 Model Tags . . . 3

2 Example Datasets **3**
- 2.1 Automatic Entry . . . 3
- 2.2 Overview . . . 4
- 2.3 Additional Vectors of the Examples . . . 4

3 Using Datasets in the Tools **5**
- 3.1 Static Model (01) . . . 5
- 3.2 Dynamic Model (02) . . . 5
- 3.3 IO Calculator (Four-Dimensional) . . . 6
- 3.4 Data Converter (03) . . . 6

4 Databox Management **6**
- 4.1 Functions at a Glance . . . 7
- 4.2 Creating a New Dataset . . . 7
- 4.3 Names of the Additional Vectors . . . 7
- 4.4 Storage Space . . . 8

5 Data Structure **8**
- 5.1 JSON Structure of a Dataset . . . 8
- 5.2 Fields . . . 9

6 Backup and Transfer **9**

7 Troubleshooting **9**
- 7.1 Databox Empty or Datasets Missing . . . 9
- 7.2 Error When Saving . . . 10
- 7.3 Import Skipped . . . 10

8 Security and Data Protection **10**

9 Frequently Asked Questions **10**

10 Quick Reference **11**

Literature **11**

About this Document

This documentation describes the databox of the English input-output tools. The databox stores IO datasets centrally in the browser; all tools access it. When first opened it already contains 26 example datasets; each dataset carries tags for the models it is intended for.

1 Introduction

1.1 What is the Databox?

The **databox** is the common data store of all input-output tools. It enables:

- **Central storage** of IO datasets (matrix A, final demand Y, final use, additional vectors, capital matrix, metadata)
- **Exchange** between the static model, the dynamic model, the IO calculator and the data converter
- **Example datasets** that are added automatically when a tool is first opened
- **Model tags** (static, dynamic, four-dimensional) for filtering and for the picker lists
- **Export and import** as JSON, export as CSV and Excel

1.2 Concept

Databox (browser localStorage)

Static model (01)	load dataset, values for the extensions
Dynamic model (02)	load dataset, values for models 1–3
IO calculator	load datasets, save results
Data converter (03)	prepare and save Destatis/Eurostat data
Databox management	view, export, import, delete

1.3 Where Are the Data Stored?

- **Location:** browser localStorage, key `ioDataboxDatasets_en`. The data are *not* transmitted to the server. The German tools use a separate databox (key `ioDataboxDatasets`), so English and German examples do not mix.
- **Separation:** each browser, each device and each private window has its own databox. The online version and locally opened files also have separate databoxes.
- **Persistence:** the data are retained until the site data ("cookies and site data") are deleted.
- **Capacity:** usually 5–10 MB; the management page shows the storage in use.
- **Transfer:** export the databox as JSON and import it on the other device.

1.4 Model Tags

The databox is one common store; datasets are tagged by model instead of being kept in separate stores. This allows datasets that fit several models (e.g. Destatis) to be used without duplication.

Tag	Set if the dataset ...
Static	contains matrix A and Y
Dynamic	contains a capital matrix B, a matrix B^I, the scalar B_I or r
Four-dimensional	contains a four-dimensional structure (field `io4d`: flows or make-use)

The tags are derived from the content; example datasets bring them along. In the management page they can be changed for each dataset with check boxes (stored in the field `modelle`). The dynamic examples and the reordered Destatis datasets are tagged only as "Dynamic", the 5×5 example only as "Static".

Picker lists in the tools: the matching datasets are at the top – first those intended only for this model, then the other matching ones, separated into examples and own datasets. Datasets for other models are collapsed under "More datasets" and remain loadable.

2 Example Datasets

2.1 Automatic Entry

When an IO tool is first opened, the example datasets are added. A new version of the examples updates existing examples automatically (data, description, source); own datasets remain unchanged. Deleted examples do not come back unasked; they can be restored with **"Restore example datasets"**.

2.2 Overview

Dataset	Dim.	Models	Final use	Additional data
Dynamic equilibrium	3, 4, 6, 8	dyn.	$Y = C$, $I = 0$	r = 1–4, B^I ($r \times r$), $X(0)$; 3×3 with capital matrix B
Dynamic 3×3 growth	3	dyn.	$Y = C + I$	$r = 1$, B^I, B, $X(0)$
Dynamic 10×10 growth	10	dyn.	$Y = C + I$	capital matrix B, $r = 1$, $B^I = 5.5$, $X(0)$
Example 5×5 open economy	5	stat.	C, I, V, E	fictitious additional vectors (all extensions), sector names
Example 3×3 to 10×10 (ZIP)	3–10	stat., dyn.	C, I; V = E = 0	fictitious additional vectors, $B_I = 5.5$, r
Example 10×10 $r = 2$ (ZIP)	10	stat., dyn.	C, I; V = E = 0	as above, partitioning $r = 2$
Germany 2022 (Destatis)	72	stat., dyn.	C, I, V, E	shares, persons employed, output, capital matrix B, $B_I = 1.887$
Germany 2022, aggregated	20, 15	stat., dyn.	C, I, V, E	as 72×72
Germany 2022, investment goods first	20, 15	dyn.	C, I, V, E	reordered, $r = 3$, B^I = block of B
ditto, B scaled	20, 15	dyn.	C, I, V, E	as above, B scaled up – **not the real data situation**
Example 1 (5 goods, 2 sectors)	10	4D	–	numerical example, $N_1 = \{1, 2, 3\}$
Example 2 (5 goods, 2 sectors)	10	4D	–	own example, $N_1 = \{1, 2, 3\}$
Example 3 and 4 (3 goods, 2 sectors)	6	4D	–	reduced example or second example
Make-Use Example 1 and 2	3	4D	–	U, V, y (3 goods, 3 sectors)

2.3 Additional Vectors of the Examples

Fictitious examples (5×5 and ZIP):

- Import shares in three groups:
  `Import_RawMaterials_Share`, `Import_IntermediateProducts_Share`, `Import_Services_Share`
- `GrossValueAdded_Share`, `Wages_Share`, `Profits_Share`, `Taxes_Share`
- `CO2_Coefficient`, `Energy_Coefficient`, `Employment_Coefficient`

For each sector the shares satisfy: column sum of A + imports + wages + profits + taxes = 1. The ZIP examples describe a closed economy ($Y = C + I$), because the dynamic model builds on them.

Dynamic examples: fictitious, from the test data for the dynamic model; start value $X(0) = (E - A)^{-1}Y$ as additional vector `X0 (start value)`, partitioning as `r (partitioning)`, investment coefficients in the field `capitalMatrixBI`. The examples 4×4 to 8×8 have no full capital matrix and are intended for model 3.

Destatis data: Federal Statistical Office of Germany (Statistisches Bundesamt), input-output accounts 2022 (revision 2024), tables 81511-03 and 81511-07; Data licence

Germany – attribution – version 2.0. Additional vectors: `GrossValueAdded_Share`, `Wages_CompensationOfEmployees_Share`, `Profits_GrossOperatingSurplus_Share`, `Taxes_OnProductionAndProducts_Share`, `Import_IntermediateInputs_Share` (shares of output), `Employment_per_EURm_Output`, `Output_EURm`. Sector names follow the English CPA product classification. The capital matrix B is illustrative and does not come from Destatis; the data are therefore only of limited suitability for the dynamic models.

Germany 2022, investment goods first: in addition to the originals, the 20×20 and 15×15 datasets are available in an order rearranged for dynamic model 3. The three sectors with the largest gross fixed capital formation come first – for 20×20 construction, research and development, machinery and vehicles (69 % of gross fixed capital formation), for 15×15 groups 7 (construction), 11 (incl. R&D) and 5 (metal, machinery) (81 %). The other sectors follow in the original order. Preset are $r = 3$ and B^I as the 3×3 block of B. Matrix A, final demand and additional vectors are only reordered; the static result remains the same.

Caution: illustrative coefficients

In the variants "B scaled (illustrative)", the capital matrix, which is illustrative anyway, is scaled up by a factor of about 1.7 (mean column sum 3, B_I accordingly). These coefficients do **not reflect the real data situation**; they only serve to illustrate that larger capital coefficients lengthen the simulation path. The stop after a few periods remains (model 2: 4–5, model 3: 4–7 periods instead of 3–5).

3 Using Datasets in the Tools

3.1 Static Model (01)

1. Input method "Databox" (default) → **"Load data from databox"**.
2. Click a dataset in the list (examples and own datasets); the list closes.
3. Shown are the source, matrix A with Y and a table with final use and all additional vectors per sector (from 10 sectors as an excerpt, complete view available).
4. **"Calculate"**. In the extensions, **"Load values"** takes over the matching vectors.

3.2 Dynamic Model (02)

1. Input method "Databox" (default) → **"Load data from databox"**; the dynamic datasets are at the top.
2. After the selection the static model is calculated, and the values are taken over for all three models: capital matrix B (otherwise $B_I \cdot E$), r, B^I, start value, given demand and $I(0)$. The display shows for each model what was taken over or is missing.
3. B^I ($r \times r$) comes from the field `capitalMatrixBI`, otherwise from the $r \times r$ block of the capital matrix B, otherwise from $B_I \cdot E$. If the dataset contains no r, r is selected in model 3; B^I is then formed immediately.
4. Given demand: C or $C + V + E$; for data with exports and inventories, $C + V + E$ is preset.
5. In each model, **"Take values from the loaded dataset"** restores the values.

3.3 IO Calculator (Four-Dimensional)

1. In tabs 1, 3, 4 and 5, **“Load data from databox”** opens a list of the matching datasets: flow tables (goods × sectors) for tabs 1, 3 and 5, make-use datasets for tab 4. After the selection the list closes and the dataset is shown; tabs 1 and 3 calculate immediately.
2. In tabs 3 and 5 the group of goods N_1 is taken from the dataset (Example 1 and 2: $N_1 = \{1, 2, 3\}$).
3. Static datasets are listed under “More” and are loaded as n goods with one sector.
4. **“Save current result in databox”** (tab 1) stores the complete structure (n, m, Z, Y) as well as A and Y; tag “Four-dimensional”.

3.4 Data Converter (03)

The converter prepares original tables and saves them in the databox:

- **Destatis:** embedded example (input-output accounts 2022, 72 production sectors, original German labels) or own Excel file; intermediate inputs, output, final use, additional rows, aggregation.
- **Eurostat:** SDMX-CSV or TSV.
- Saving with **“Save in databox”**; source and sector names are saved as well.

Note: dimensions

All matrices and vectors of a dataset must match the dimension n: A and B are $n \times n$, all vectors have n values. Scalars such as B_I and r are stored as a vector with one value.

4 Databox Management

The management page (`databox-system_en.html`) can be reached via the navigation of all IO tools.

4.1 Functions at a Glance

Button	Function
Filter All / Static / Dynamic / Four-dimensional	shows only datasets with this tag; export and Excel refer to the selection shown
Export databox (JSON)	save the datasets of the selection in a JSON file
All datasets as Excel	one Excel file of the selection: overview (with models) plus, for each dataset, the sheets Info, Matrix A, Vectors, Capital matrix, B^I
Import databox (JSON)	add datasets from a JSON file; existing ones (same name and dimension) are skipped
Restore example datasets	add missing examples again
Delete all	empty the databox (with confirmation)
Per dataset:	
Models (check boxes)	change the tags of the dataset
Show info/readme	description and source
CSV: matrix A	matrix A with sector names (semicolon, decimal point)
CSV: vectors	Y, final use and all additional vectors per sector
CSV: capital matrix	capital matrix B (if available)
CSV: B^I	investment coefficients B^I ($r \times r$, if available)
CSV: IO calculator	four-dimensional data in the format of the IO calculator (Z, Y or U, V, y), can be read in again
Excel: all data	one Excel file with Info (incl. scalars such as B_I), Matrix A, Vectors, Capital matrix, B^I or Z (4D) or U/V (make-use)
Delete	remove the dataset (with confirmation)

4.2 Creating a New Dataset

1. Section **"Add new dataset"**: enter a name.
2. Enter matrix A (one row per sector) or load it as a CSV file.
3. Enter vector Y (in one row or listed vertically) or load it as CSV.
4. Optionally **"+ Add additional vector"**: name, type (final use component or additional vector) and values.
5. **"Save dataset"**.

CSV format: comma, semicolon or tab; with semicolon or tab a decimal comma is also allowed (German Excel export). A header row and sector names in the first column are skipped; empty cells and "-" count as 0. Example:

```
Sector,S1,S2,S3
S1,0.15,0.20,0.10
S2,0.25,0.10,0.15
S3,0.10,0.15,0.20
```

4.3 Names of the Additional Vectors

The tools recognise vectors by their name (case-insensitive). Recommended names:

Use	Name contains
Final use	`C`, `I`, `V`, `E` (as final use component)
Import groups (01, step 2)	"Import"
Cost structure (01, step 3)	"Wage", "Profit", "Tax"; imports = sum of all import vectors
Price model (01, step 4)	"ValueAdded" or "value added"
Satellite accounts (01, step 5)	"CO2", "Energy", "Emission", "Environmental", "Employment"
Dynamic model (02)	`B_I`, `r (partitioning)`, `X0 (start value)`; capital matrix B and B^I as separate fields

German names (e.g. from the original Destatis sheets in the converter) are also recognised.

4.4 Storage Space

The management page shows the storage in use. If the storage is full, a message appears; then delete datasets no longer needed or export them first. Guide values: small examples approx. 2–3 KB each, 20×20 approx. 13 KB, 72×72 with capital matrix approx. 80 KB; all 26 examples together occupy about 350 KB.

5 Data Structure

5.1 JSON Structure of a Dataset

```
{
  "id": 1793732382331,
  "name": "Example 3x3 (ZIP dataset)",
  "dimension": 3,
  "created": "11/4/2026, 7:45:12 PM",
  "source": "Example data (ZIP dataset)",
  "readme": "EXAMPLE DATASETS ...",
  "isExample": true,
  "modelle": ["statisch", "dynamisch"],
  "matrixA": [[0.15,0.20,0.10],[0.25,0.10,0.15],[0.10,0.15,0.20]],
  "vectorY": [130, 170, 135],
  "finalDemandComponents": {
    "C": [100, 150, 120], "I": [30, 20, 15],
    "V": [0, 0, 0],       "E": [0, 0, 0]
  },
  "additionalVectors": {
    "Import_RawMaterials_Share": [0.0245, 0.0352, 0.0446],
    "Wages_Share": [0.2494, 0.2864, 0.2977],
    ...
    "B_I": [5.5],
    "r (partitioning)": [1]
  }
}
```

The internal tag values `statisch`, `dynamisch` and `vierdim` are the same in both language versions.

Optional fields:

- `sectorNames`, `capitalMatrix` (e.g. Destatis),
- `capitalMatrixBI` (dynamic examples).

5.2 Fields

Field	Description
`id`	unique ID (timestamp)
`name`	name of the dataset
`dimension`	size n
`created`	creation date
`source`	origin or source reference
`readme`	description (optional)
`isExample`	`true` for example datasets
`modelle`	model tags, e.g. `["statisch", "dynamisch"]` (optional; otherwise derived)
`matrixA`	technical coefficients ($n \times n$)
`vectorY`	final demand (n values)
`sectorNames`	sector names (optional)
`finalDemandComponents`	final use, e.g. C, I, V, E
`additionalVectors`	additional vectors (n values) and scalars (1 value)
`capitalMatrix`	capital matrix B ($n \times n$, optional)
`capitalMatrixBI`	investment coefficients B^I ($r \times r$, model 3, optional)
`io4d`	four-dimensional data (optional): `{typ: "verflechtung", n, m, Z, Y, N1}` (flows) or `{typ: "makeuse", n, m, U, V, y}`

6 Backup and Transfer

Recommended backup

After important changes or before deleting browser data:

1. Open the databox management → **“Export databox (JSON)”**.
2. Store the file safely, e.g. `databox_YYYY-MM-DD.json`.
3. To restore or transfer to another device: **“Import databox (JSON)”**.

For further processing in spreadsheets, use the Excel export (all datasets or individual ones). The CSV files can be loaded directly back into the tools.

7 Troubleshooting

7.1 Databox Empty or Datasets Missing

1. **Different browser or device:** each databox is separate; transfer datasets via JSON.
2. **Private window:** the databox is discarded when the window is closed.
3. **Browser data deleted:** import from a JSON backup; restore examples with “Restore example datasets”.
4. **Online and local:** the online version and locally opened files have separate databoxes.
5. **English and German tools:** they use separate databoxes; transfer datasets via JSON export and import if required.
6. **Display outdated:** “Refresh list” or reload the page (F5).

7.2 Error When Saving

- *"The databox is full"*: delete datasets or export them first.
- *"The databox could not be saved"*: browser storage disabled (e.g. private window or privacy settings).
- Matrix not square or vector of wrong length: check dimensions.
- Values read incorrectly: check separator and decimal sign (see CSV format).

7.3 Import Skipped

In the JSON import, datasets with the same name and the same dimension are skipped. To replace one, delete or rename the existing dataset first.

8 Security and Data Protection

- All data remain locally in the browser; nothing is transmitted to a server.
- The data are not encrypted; anyone with access to the browser's user account can see them.
- For confidential data: use only on your own devices, delete after use, store backups securely.

Licence note: when passing on datasets based on official data, keep the source reference (field `source`); for Destatis the Data licence Germany – attribution – version 2.0 applies.

9 Frequently Asked Questions

Is the databox empty on the first visit?
No. When an IO tool is first opened, the example datasets are added.

Can I use the databox on several devices?
Only via export and import (JSON).

Are data lost when I close the browser?
No, only when the browser data are deleted, in a private window or by deleting in the databox.

Do I have to separate the databox by models?
No. There is one databox; the tags make sure that each tool shows the matching datasets at the top.

Are my own datasets overwritten by an update?
No. Only the example datasets are updated.

Can I pass datasets on?
Yes, as JSON (for import) or as Excel/CSV.

10 Quick Reference

Action	Procedure
Load	static and dynamic model, IO calculator (tabs 1, 3, 4, 5) → "Load data from databox"; converter → databox section
Save	data converter or IO calculator → "Save in databox"; management → "Add new dataset"
View	static or dynamic model (tables) or management (info, CSV, Excel)
Filter	management → All / Static / Dynamic / Four-dimensional
Back up	management → "Export databox (JSON)"
Transfer	export JSON → import on another device
Restore examples	management → "Restore example datasets"
Delete	management → "Delete" or "Delete all"

Literature

P. Fleissner, W. Böhme, H.-U. Brautzsch, J. Höhne, J. Siassi, K. Stark: *Input-Output-Analyse. Eine Einführung in Theorie und Anwendungen.* Springer-Verlag, Vienna, New York 1993.