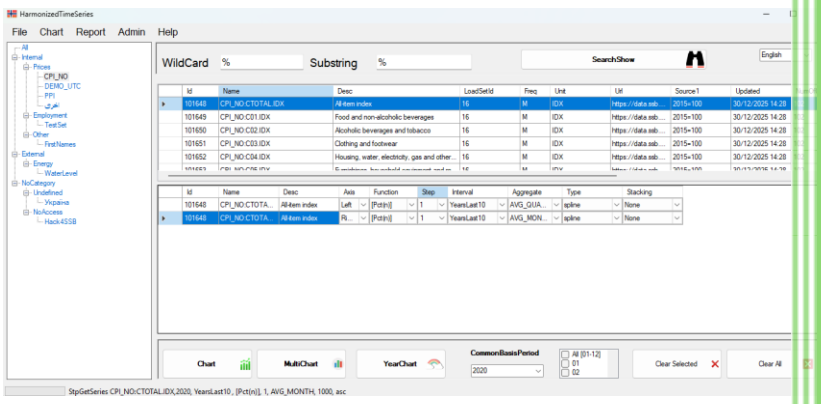
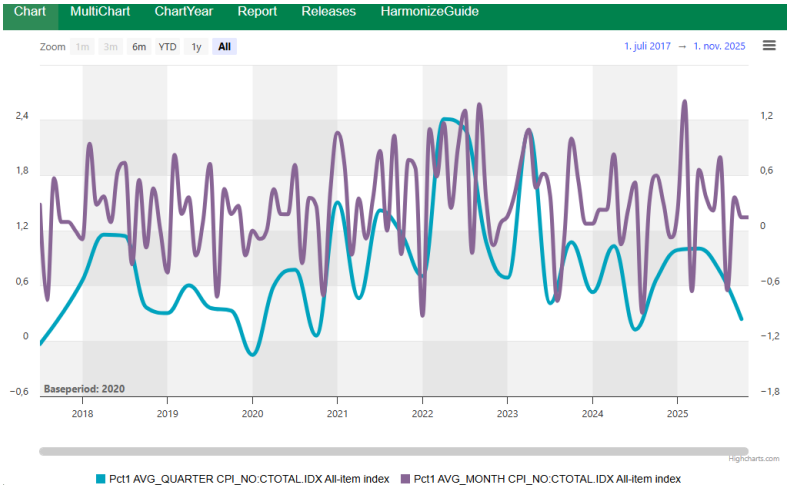
Harmonize structured timeseries

2026

Timeseries tool & database with api

Special edition. With demo database. Contains export/import samples using Python.





Erik Soeberg

rsb@ssb.no

11/01/2026

Harmonize User and Installation Guide 2026

Summary

Harmonize is designed to make time series data accessible to users **without programming skills**, while also helping management and analysts save time. The system stores time series in a generic, flexible format, allowing **easy comparison of datasets with different frequencies**. The system is based on MS SQL-Server database. The system also allows users to access the database-procedures (**api**) using tools like **Python** and **R**. By using the Harmonize application, the time series is easily available for many users, having different access, for comparison, with different functions, aggregates and the use of common base year.

Examples:
- Chart a **quarterly index** with a **monthly index** seamlessly.
- Adding a **weekly, monthly, quarterly and annual series** correctly; - Charts can use a **common base year** or **base period**, even if datasets have different base years (e.g., 2015=100 vs 2020=100).

Charts are **zoomable and scrollable** because all data uses datetime formats. You can apply **aggregations** like `avg_quarter` or `avg_year`, and **functions** such as *diff_n*, `pct_1`, `pct_2`, ... `pct_n` (where `n` is the lag).

Harmonize can be run via the **C# application for windows**, using Highcharts to make modern charts. The Html scripts can be modified to use your colors / fonts and css information, you can also replace higcharts.com with other similar open source javascript charting tools, which are not recommended.

Harmonize can save and retrieve chart selections, for reuse with fresh data later. The dynamic intervals will make sure current year is the current year. However static intervals may be applied if users always want a chart to be with observation in 2025, always.

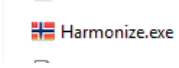
Contents

Installation Guide

Step 1 — Application Installation

Requirements: Google Chrome or any modern browser.

- Download files, unzip the files in a folder C:\Harmonize\app\, and Start

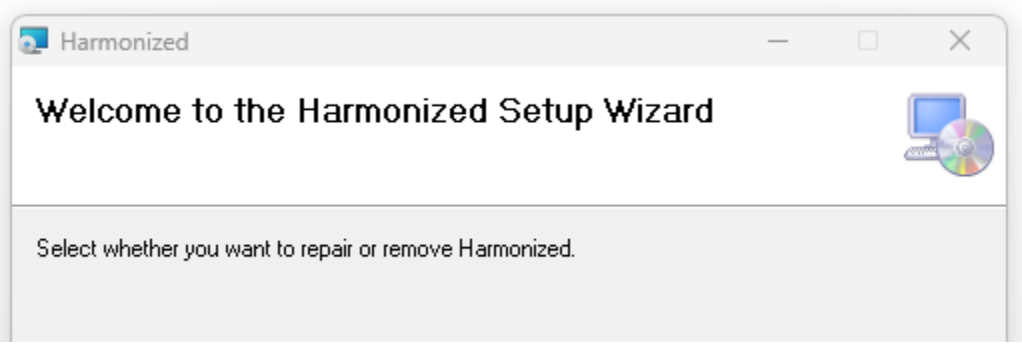
  Harmonize.exe (with a flag) 

  The Harmonize system will initially access a Demo database, with read access to some demo data. When you want to use Harmonize with your own custom data you need your own MSSQL database that only you and your company can access.

- Alternatively, the more professional alternative, however this required admin rights: Run `Setup.exe` to copy files to your chosen folder. This requires you to download the Setup..
  - Default location:
    `C:\Harmonize\Common\`

- Main Files installed: (in addition there are some python samples to import & export)

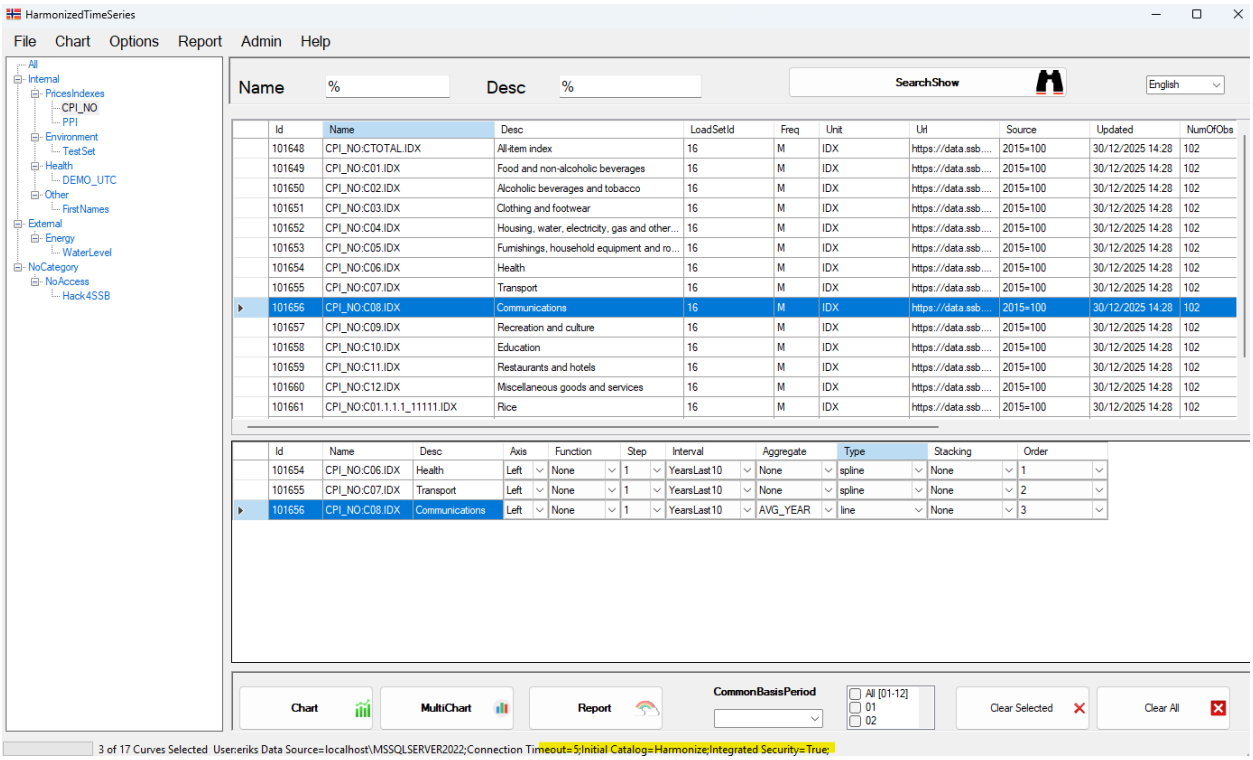
Harmonize.exe	Main application
Harmonize.css	Styling for templates
Harmonize.bak	Database backup, to be restored in your mssql environment.
Chart.html	Single chart template
MultiChart.html	multi-chart template
YearChart.html	Yearly overlay chart template, using current year intervals
Report.html	Report template for direct repoting
Release.html	Release notes
connection.txt	Sample database connection (username/password)
Menues.txt	Menu labels (multi-language support)
ffi.py	generic py script that can import csv files to datasets in the sql
FFItestfile.csv	Sample dummy csv file, to be imported by the **ffi.py** program
cpi.py	Sample reading CPI from *statbank* Norway json-stat format
ppi.py	Sample reading ppi-index from *statbank* in Norway
GetCurvetoPython.py	Gets series from the database and chart it



The application can be installed to a **shared folder** (read access) for multiple users, together with the database connection.txt and the html templates. If usernames and passwords are hardcoded in connection.txt, all files need to be placed at each user personal home drive. The best set up is to us a shared folder, using a trusted database connection.txt not containing any passwords,

Personal data, temporary datafile will be stored by default at `C:\Harmonize\Chart,` can be changed in connection.txt, but must be a folder that is private for each user.

You may initially try to double click at the file ***Releases.html,*** and verify it opens in Chrome or Edge. If your html-files is defined to open in ie. *notepad* or any other non-browser application, this needs to be changed, so a browser like Chrome or Edge automatically opens ***html*** files. When Harmonize application creates charts, they will open in your default browser and read the freshest MyData.js available in your personal C:\Harmonize\Chart folder. To set default browser use Windows+I



When the application is launched, the connection string is shown in the bottom form. In the example above it's used a trusted connection, to user that only has public access to the Harmonize sql database.

By double clicking (selection) one or more series, they will jump to the bottom grid in the application, where you can select interval, function, aggregation, base-year, and chart style. When you have one or more series in the bottom grid, the Chart buttons will be enabled. Inside the charts, you can see the charts as a report, However, you can also report data directly from the application, from Report Menu in the top application bar.

To utilize Harmonize, you need a **Microsoft SQL Server database** that your / your company controls with your data series.

- Install SQL Server on your machine or server.
    - You may use **SQL Server Express** if a full version is not available.

- Restore the Harmonize database backup ***harmonize.bak*** with the system. Logg in using an administrator like *sa*, use *master*, and restore the database file. You can call the mssql -database Harmonize, but other names can be used.

- Verify that you can **connect to the database** using SQL Server Management Studio (SSMS).
    - Ensure that the **user credentials** or **trusted connection** are working.

    - Harmonize is default installed with a trusted connection in *connection.txt*

- Note the database name and connection details; you will need them when configuring ***`connection.txt`***. See the sample `connection details` strings in bottom of *connection.txt*. Try to use an integrated / trusted connection, making connection details safer and more secure. The first line only, in the *connection.txt* holds the connection details. Line #2 is the name of the directory where the users temporary data for charting and reporting is stored. Line #3 are the languages in your Menue.txt file that will be selectable in the application. Line# is the default Language, a 0 is the first language: English. Line #5 is the maximum number of rows returned for one series, This is typically only important for time series of high frequency such as hourly, minutely or secondly.

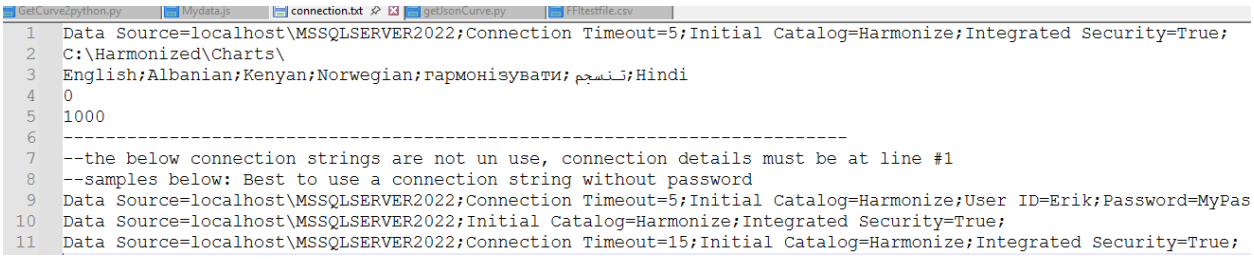
- Database Users: By design the system is to be used by ***api***, via python/R or the C# visual studio Harmonixe.exe application. The users do not need other than public access to the database, and it should be enough to run all import and extract procedures no matter if you are using python or R or Harmonize.exe
- By doing this, we ensure no bad queries are executed, and the system will perform fast and smoothly, and metadata will be updated according to design (ie updated date)

Step 2 — Database Connection Configuration

- Edit *connection.txt*. The first line must be connection string to your SQL Server database with a trusted connection not requiring database username and password. If using a trusted connection, all users can use the Harmonize files from the same shared folder. This is the best and safest set-up, and it will be simpler to upgrade to later versions. The file must be named connection.txt. Personal user ids and password are also OK, but then the application need to installed locally for each user.

- Choose where **local chart data** will be stored (`MyData.js`). Default folder is: **C:/Harmonized/Charts/** - if this is changed this path needs to be changed in the 3 html files: Chart, MultiChart, and YearChart, and Report.html. Initially let the directory and files be unchanged until you have Harmonize up and running.

- Recommendation: use connection mode as shown in *connection.txt* for **trusted connections**.



Step 3 — Using the Application

- Run **Harmonize.exe, a shortcut** should be available on your desktop**.**

- Verify that the application, list a tree with datasets:
- **Test Help, Release** from top menu to verify that the application can open your default browser, and display the data. If this does not work It will not work to show charts either.

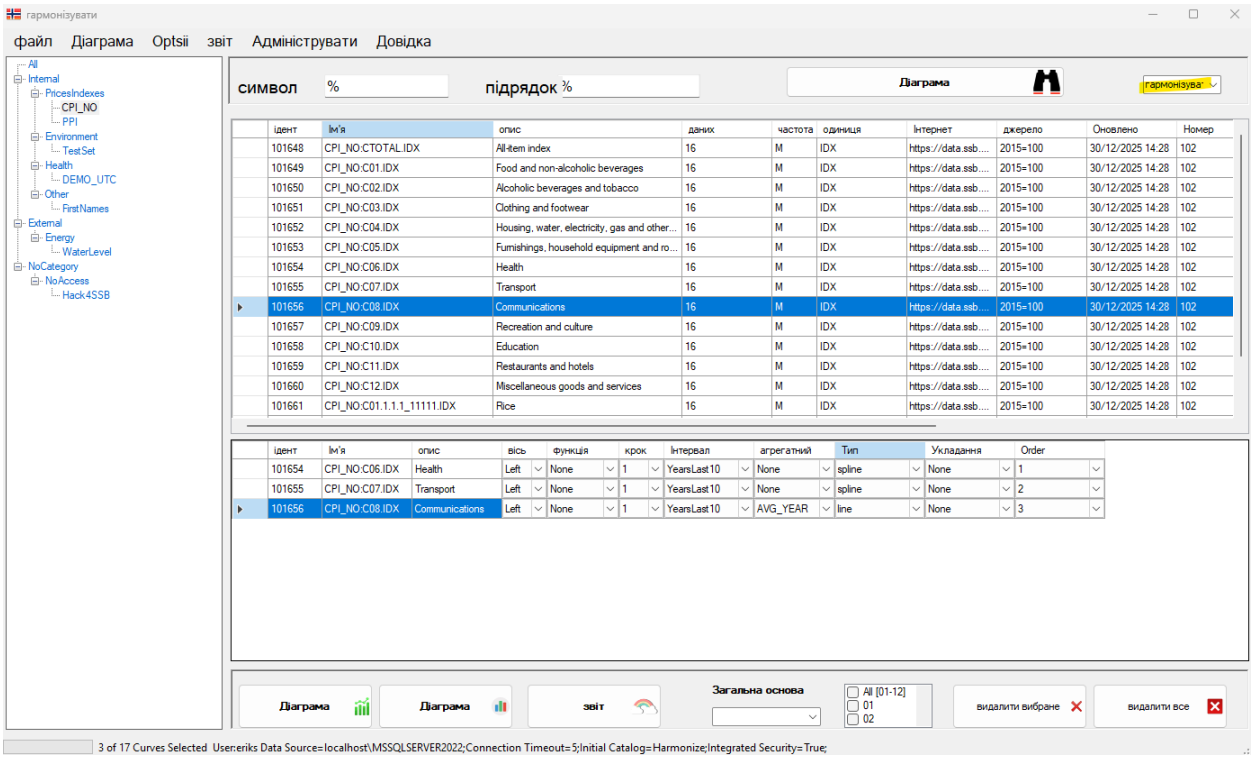
- Generate charts & Reports using:
    - **Chart**, **MultiChart**, **YearChart** is under the Chartmenue
    - **User Report button for quick horizontal & vertical reports**
    - **Report Tables also available from the inside the Charts**

- Charts are rendered in the default **browser** (Chrome recommended but Harmonize uses your default browser. You need to have a browser as a default application to open html files.)

- Modify templates as needed: (not needed but possible to customize)
    - Change fonts, colors, legends, tooltips
    - Can modify Harmonize.css so it fits your standards.
    - Use Highcharts or another compatible library

- The C# application overwrites **`MyData.js`** each time charts are generated. Copy it elsewhere if you need to preserve data. If so, you may need to copy one of the templates Chart.html to point to your local files. You may create your own html templates. It's also possible to change and customize all the html files.

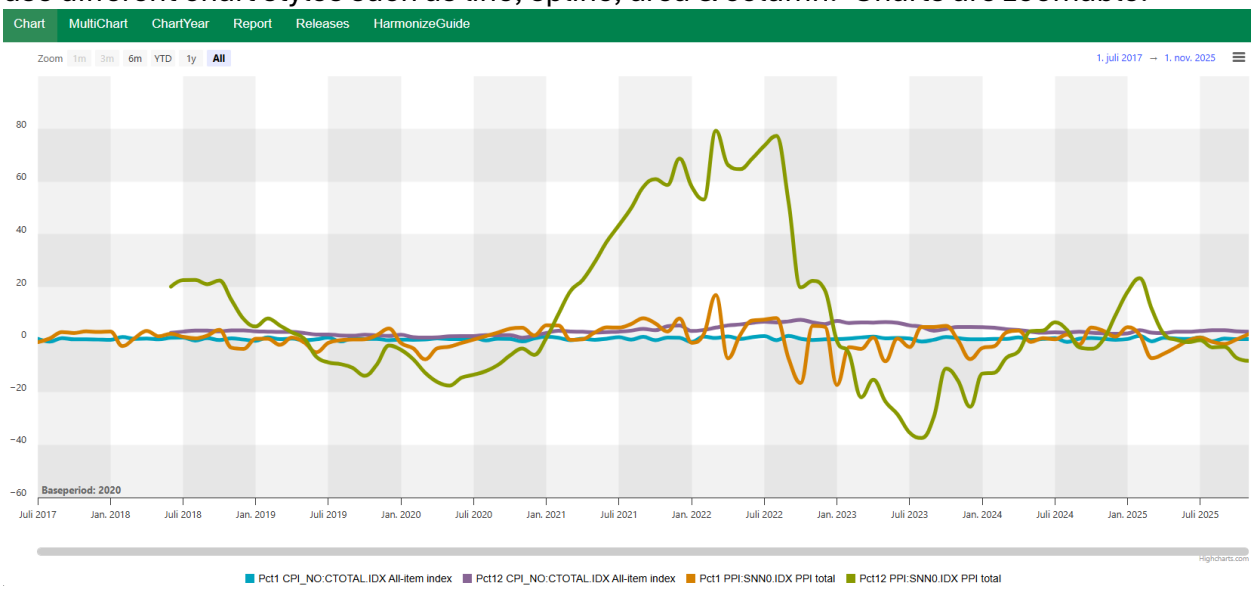
- **Important: YearChart template**: only use intervals containing data up to one year (e.g., CurrentYear-1, CurrentYear-2) More Intervals can be added into the database.

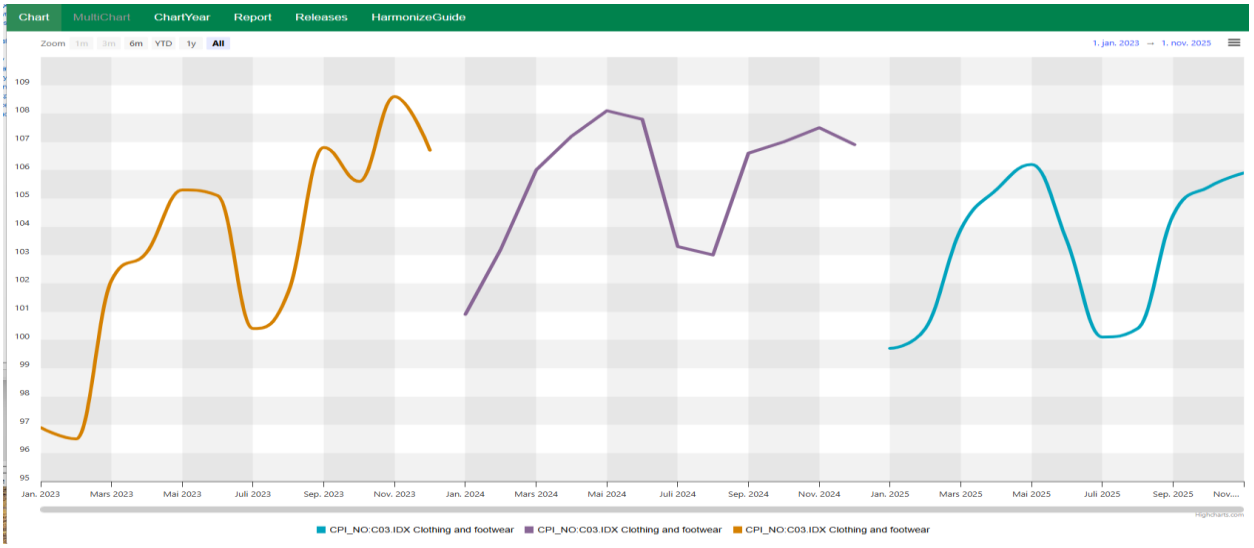

гармонізувати
файл
Діаграма
Optsii
звіт
Адмініструвати
Довідка
All
Internal
PricesIndexes
CPI_NO
PPI
Environment
TestSet
Health
DEMO_UTC
Other
FirstNames
External
Energy
WaterLevel
NoCategory
NoAccess
Hack4SSB
символ
%
підрядок %
Діаграма
гармонізувати
ідент
Ім'я
опис
даних
частота
одиниця
Інтернет
джерело
Оновлено
Номер
101648 CPI_NO:CTOTAL.IDX All-item index 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101649 CPI_NO:C01.IDX Food and non-alcoholic beverages 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101650 CPI_NO:C02.IDX Alcoholic beverages and tobacco 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101651 CPI_NO:C03.IDX Clothing and footwear 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101652 CPI_NO:C04.IDX Housing, water, electricity, gas and other... 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101653 CPI_NO:C05.IDX Furnishings, household equipment and ro... 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101654 CPI_NO:C06.IDX Health 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101655 CPI_NO:C07.IDX Transport 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101656 CPI_NO:C08.IDX Communications 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101657 CPI_NO:C09.IDX Recreation and culture 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101658 CPI_NO:C10.IDX Education 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101659 CPI_NO:C11.IDX Restaurants and hotels 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101660 CPI_NO:C12.IDX Miscellaneous goods and services 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
101661 CPI_NO:C01.1.1.1_11111.IDX Rice 16 M IDX https://data.ssb.... 2015=100 30/12/2025 14:28 102
ідент
Ім'я
опис
вісь
функція
крок
Інтервал
агрегатний
Тип
Укладання
Order
101654 CPI_NO:C06.IDX Health Left None 1 YearsLast10 None spline None 1
101655 CPI_NO:C07.IDX Transport Left None 1 YearsLast10 None spline None 2
101656 CPI_NO:C08.IDX Communications Left None 1 YearsLast10 AVG_YEAR line None 3
Діаграма
Діаграма
звіт
Загальна основа
All [01-12]
01
02
видалити вибране
видалити все
3 of 17 Curves Selected User:eriks Data Source=localhost\MSSQLSERVER2022;Connection Timeout=5;Initial Catalog=Harmonize;Integrated Security=True;

Step 4 — Template for the Chart & Report options

Chart.html (Template for n series in 1 container)

The main template takes all your selected series and chart them in one container. To compare more series, you may choose various functions and aggregations, as well as common base year functionality. Series may be stored in different datasets, and you may use different chart styles such as line, spline, area & column. Charts are zoomable.

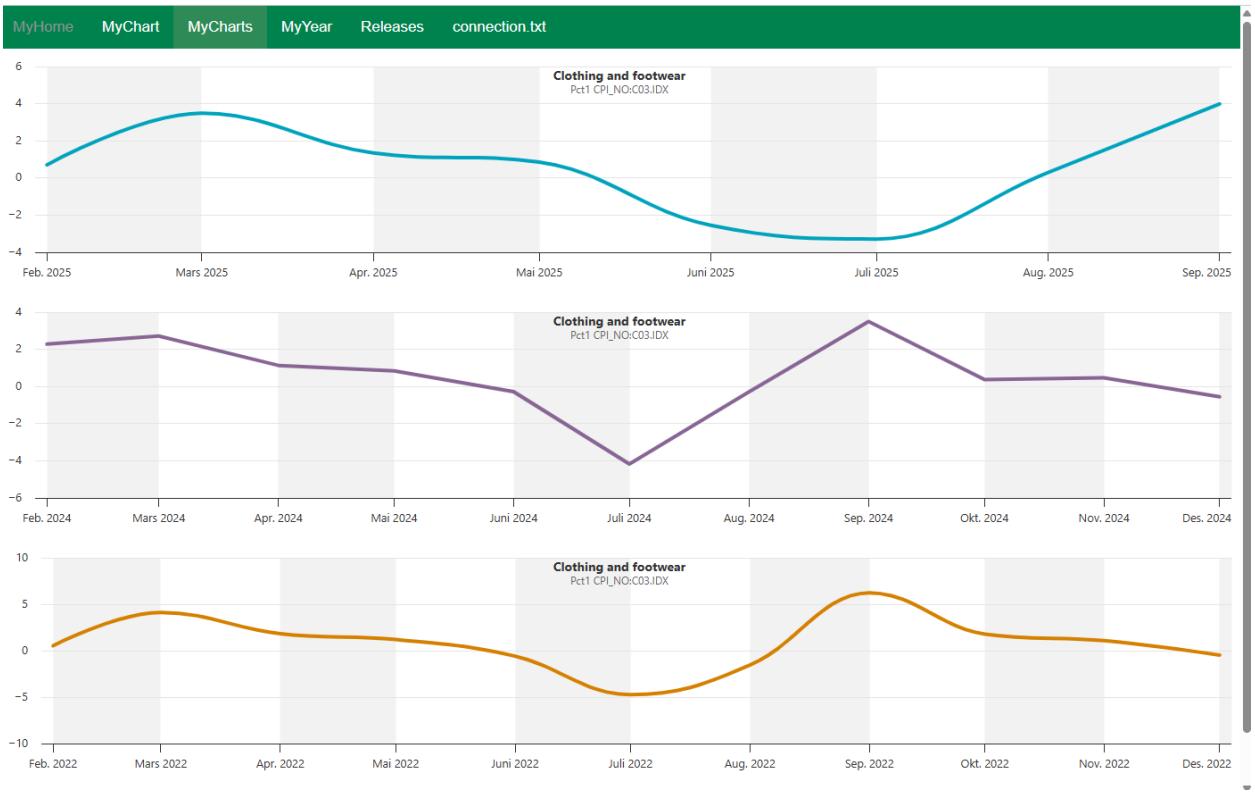




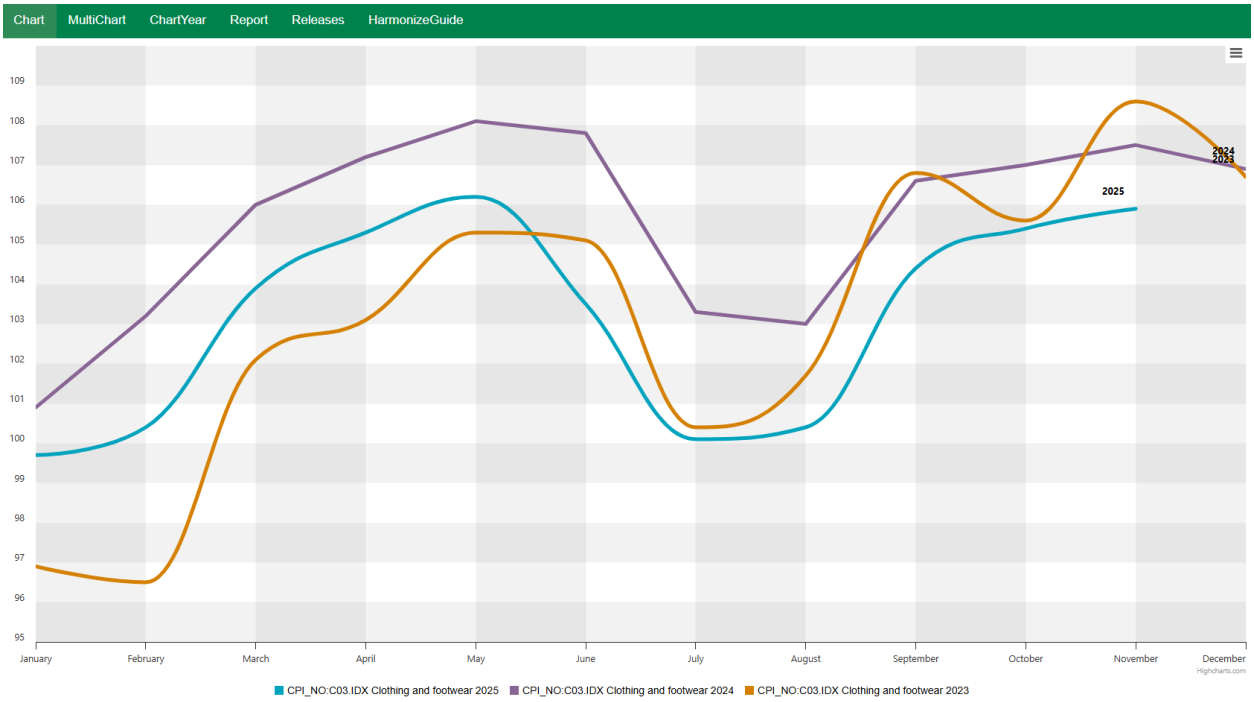
MultiChart.html (Template for n charts in separate containers)

MultiChart, will chart n series in ***n*** containers, you will see approximately 3 charts on each page, scroll down if more charts are available. The n charts are independent and not synchronized based on the x-axis.

Currently there are no print or view data as table for this template, this can easily be added in the MultiChart.html



YearChart.html – overlay (all Series need to use intervals containing one year)



The ChartYear template is only usable when used as above and correct intervals are used. A year label is printed to the labels and legends to more easily differ among the different years.

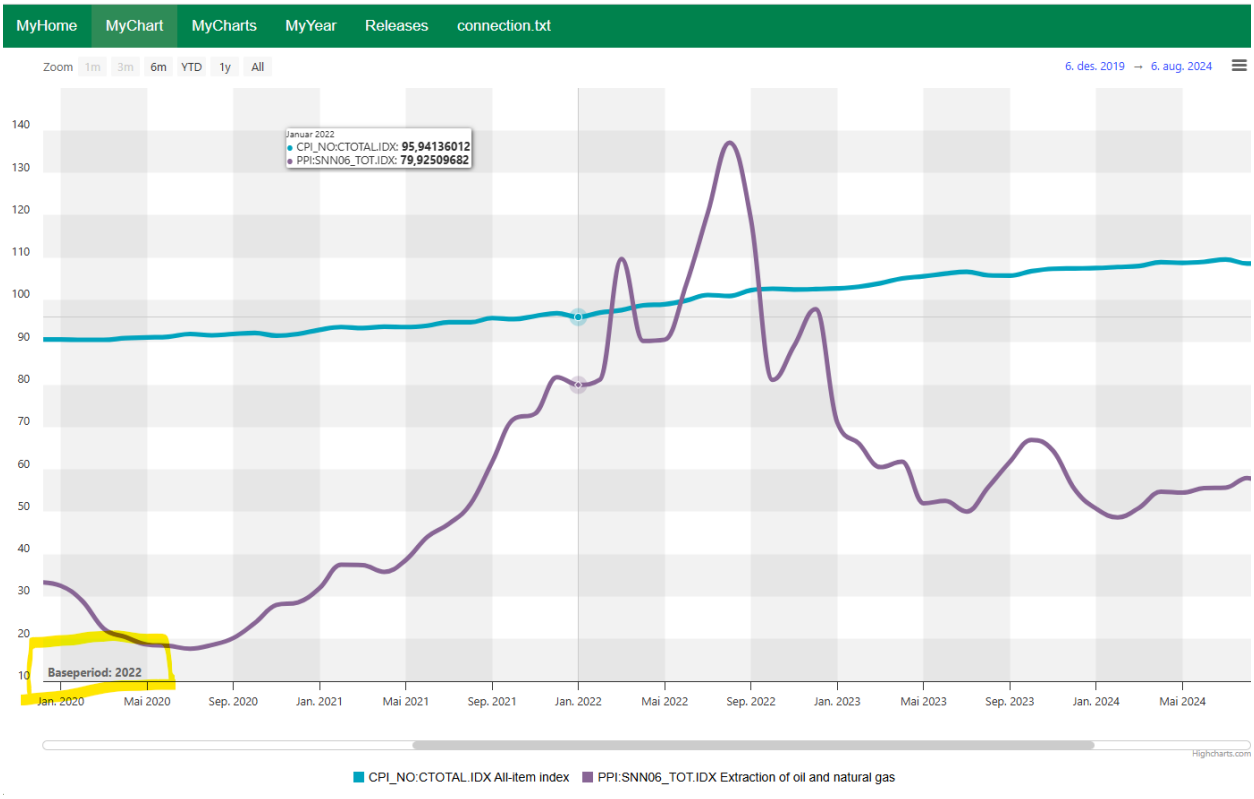
All charts can be viewed as a table, or saved as csv, se the small hamburger menu located in the top right corner in your charts.

Month	CPI_NO:C03.IDX 2025	CPI_NO:C03.IDX 2024	CPI_NO:C03.IDX 2022
January	99.7	100.9	94.4
February	100.4	103.2	94.9
March	103.9	106	98.8
April	105.3	107.2	100.6
May	106.2	108.1	101.8
June	103.5	107.8	101.2
July	100.1	103.3	96.4
August	100.4	103	94.9
September	104.4	106.6	100.8
October		107	102.6
November		107.5	103.7
December		106.9	103.2

CPI_NO:C03.IDX Clothing and footwear 2025 CPI_NO:C03.IDX Clothing and footwear 2024 CPI_NO:C03.IDX Clothing and footwear 2022

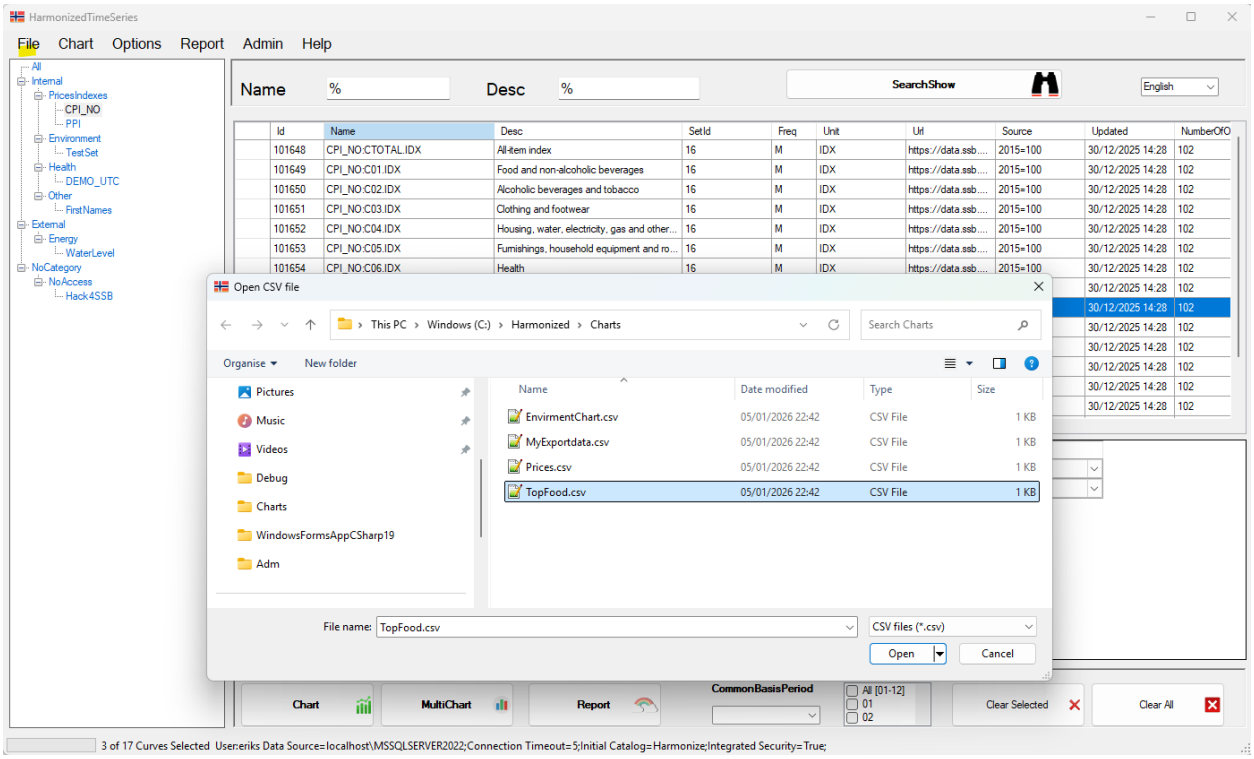
Common base year

When common base period is used, this will be printed in your charts, be aware common base year can be combined with function and aggregates. The year selected as the common year need to have data observation(s). The sample below compared CPI and PPI, 2 indexes that have different base year, now using common base year 2022 = 100.



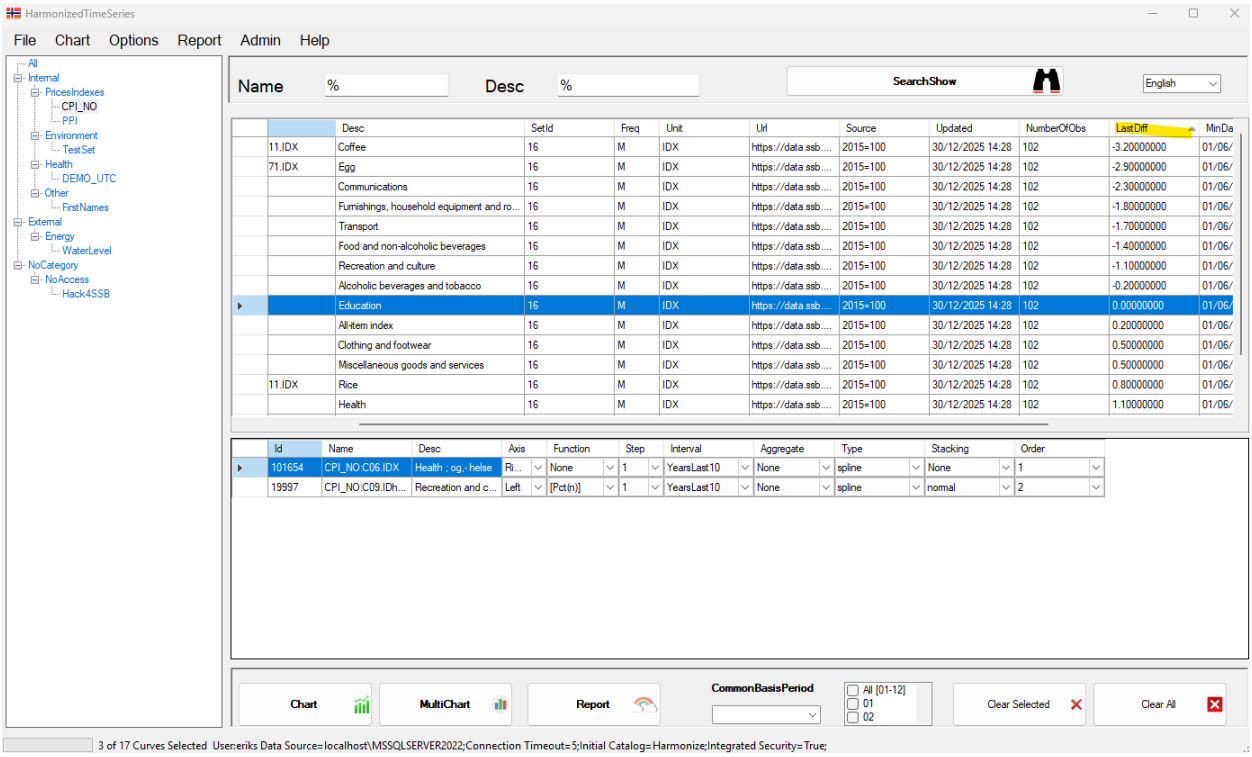
Application tricks & techniques

- When searching for series you can hold shift key down and select several series in one go. The series will then move to the selected area, and the Chart /Report options will be enabled.
- In the selected area, you can right click, and apply interval, function etc to all series
- The order column is used for sorting; by clicking on the column headers, you may sort on any column, or you can specify an order number to sort on.
- Hidden under the File menu, you can save your chart selection, with intervals, functions and all the information in the selected area**. Save** the selection.csv on your computer to recreate the chart with **fresh** data, at later point in time.
- Be aware of changes in the series names or change of access to the data since you saved your definition may cause some issues.
- # of Decimals and date formats can be selected under the Options menu.
- Shortcuts, Harmonize standard shortcuts are Alt+C => Chart, Alt+M => Multichart, Alt+R => Report to execute a Charts /Report. The application will always extract the newest data from the database before displaying it.

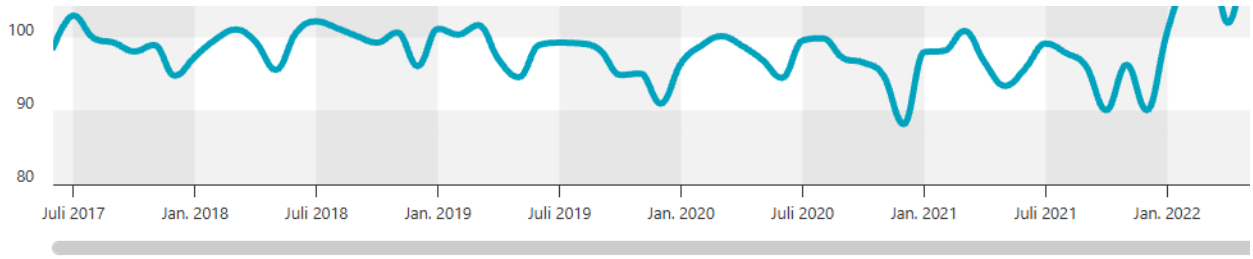


Sorting and Ordering

In the top datagrid you may sort the series in the dataset on all columns, observe that the column Last Diff sort the dataset, based on which curves had the most / least change according to the previous observation.

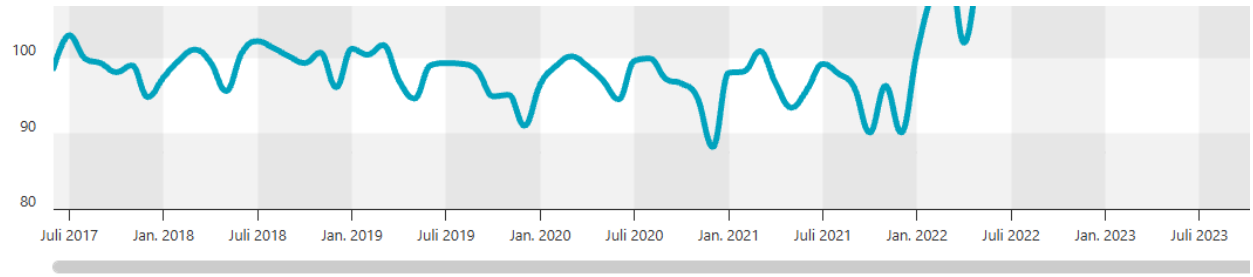


When viewing as table, clicking at Header you easily find max and min point for the series. (This is why the Date is not sorted: but the max value is September 2025.



DateTime	CPI_NO:C01.2.1.1_12111.IDX
2025 09 01 00:00	170
2025 07 01 00:00	169
2025 10 01 00:00	163.8
2025 11 01 00:00	160.6
2025 06 01 00:00	158.7
2025 08 01 00:00	155.4
2025 05 01 00:00	147.2
2025 01 01 00:00	139.6

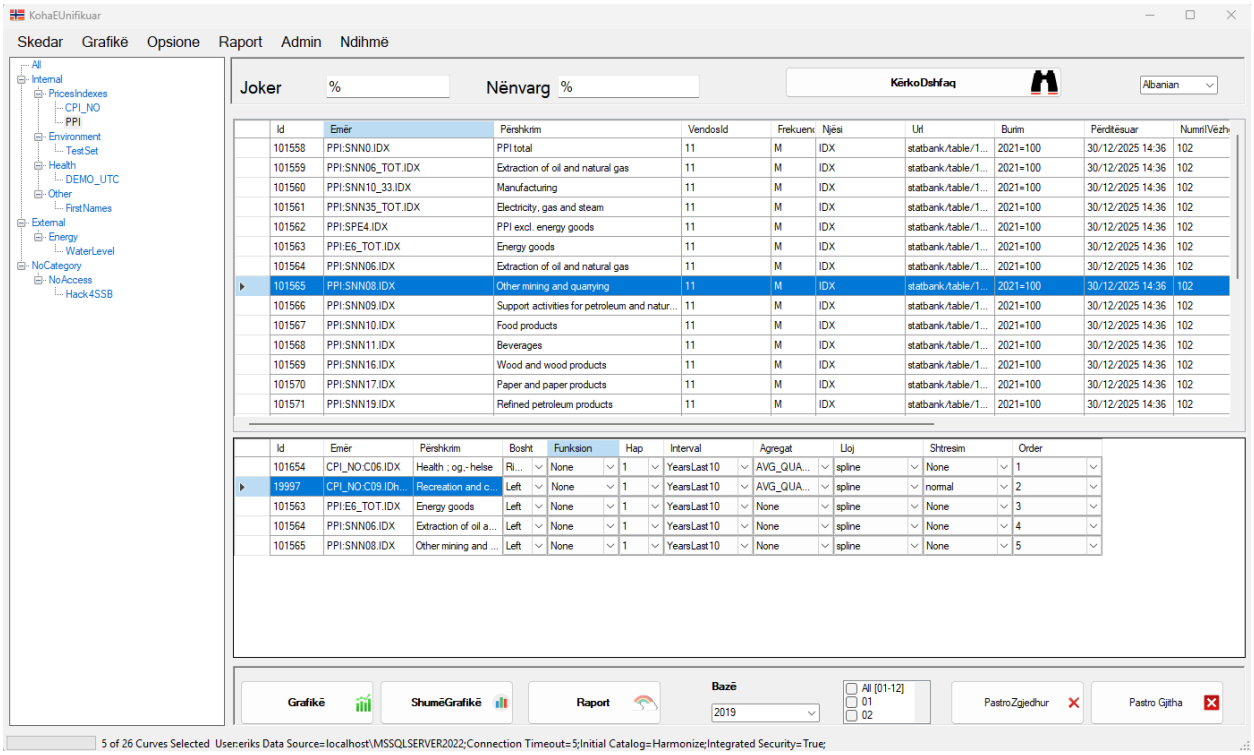
Min point:



DateTime	CPI_NO:C01.2.1.1_12111.IDX
2020 12 01 00:00	88.2
2021 10 01 00:00	90.1
2021 12 01 00:00	90.1
2019 12 01 00:00	91
2021 05 01 00:00	93.4
2020 06 01 00:00	94.5
2019 05 01 00:00	94.6
2017 12 01 00:00	94.8
2020 11 01 00:00	94.8
2019 10 01 00:00	94.9

Menus and Languages

- Edit ***Menues.txt*** for custom language labels.
- **Do not remove lines or change the order in Menues.txt**. Backup the original file before editing.



Data management

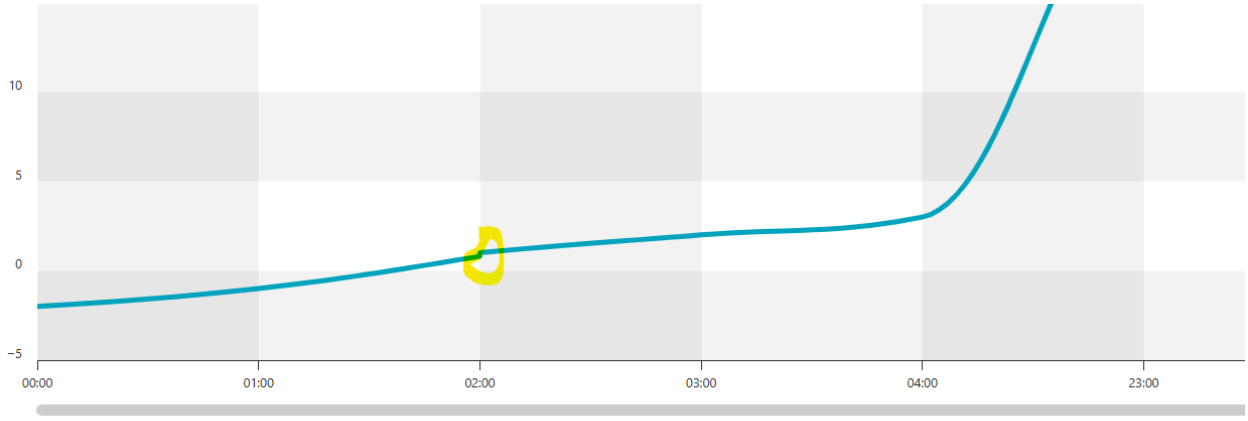
- Harmonize calls **SQL stored procedures** to generate `MyData.js`.
- Sample Python scripts are included for importing or updating data.
- Units are standardized via **Unit IDs**, editable in the *Admin* form or directly in the database.
- Interval may be added to the View inside the database, sql logic. The procedures using the intervals will query data that is **GE start_date and LT end_date**
- Additional datasets should be defined from admin, they can also be put in a tree, which is displayed leftmost inside the application.
- Utility procedures are included for deleting data and creating new datasets.
- The time series may be stored in many different tables, the data tables need to be in the same format and contain a column *value_date* in date or datetime. Choose date if you do not have hourly data.

Timezones and day light saving

When having higher frequencies such as hourly, and need to manage summer and wintertime, the system can handle this. Data needs to be stored in UTC and in the Loadset table, the dataset needs to be configured to be displayed in a different time zone than UTC. Be aware that time zones are more complicated than just visualizing the data. When you do aggregation, the system converts time zone, before applying functions and aggregations such as average day.

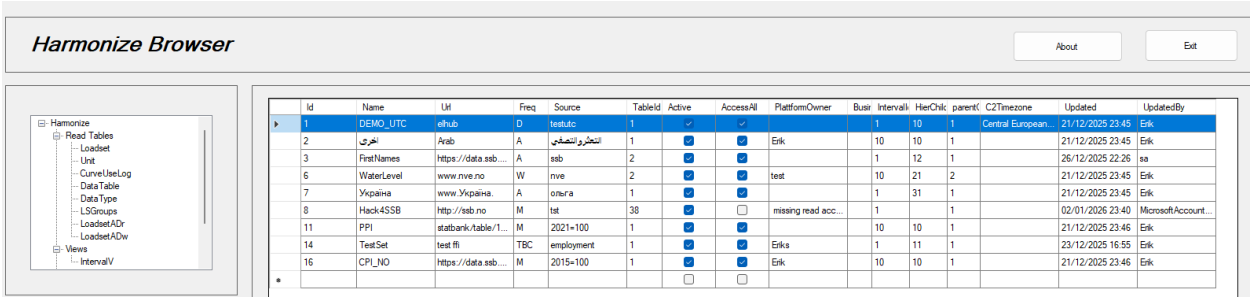
Ie hourly spot prices will have 25 observations (hours) when shifting to wintertime, but only 23 hours (see next page) when shifting from winter to summertime.

The chart below displays 2 observations on the hour 02:00, because the CET clock is set back by one hour once a year.



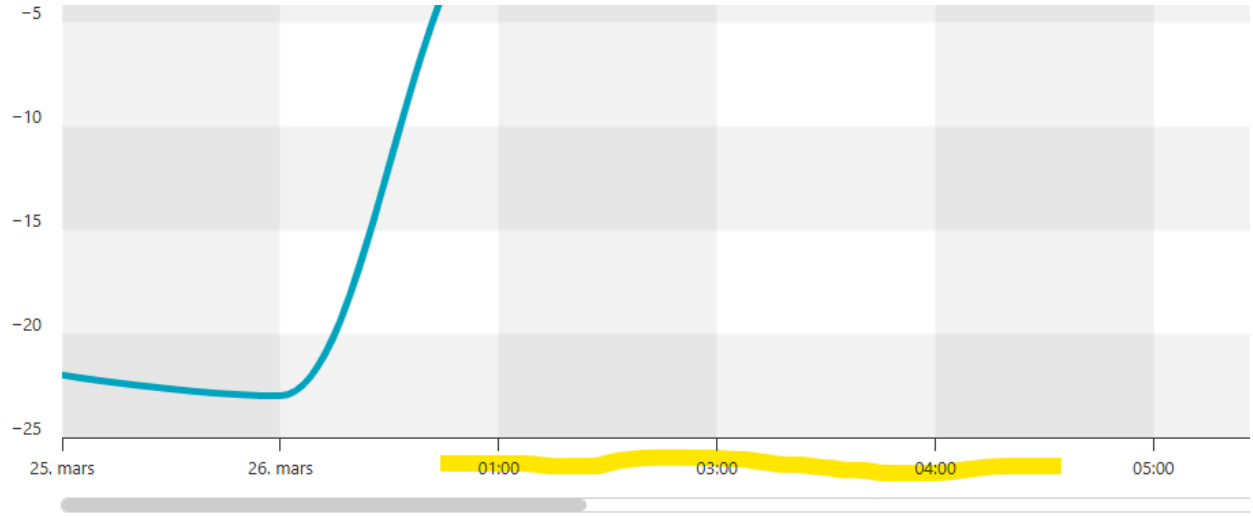
DateTime	DEMO_UTC:SOMMERVINTER
2023 10 29 00:00	-2
2023 10 29 01:00	-1
2023 10 29 02:00	0.8
2023 10 29 02:00	1
2023 10 29 03:00	2
2023 10 29 04:00	3

In the dataset configuration the dataset above is stored in UTC, but the series from this dataset is displayed in Central European time.



When having hourly data, the report information may be confusing as the standard report template use a date format year month day, but this can easily be changed to contain hours and or minutes or seconds as well.

Below, observe the gap x-axis when chart/reporting an hourly series in the winter to summer shift.



DateTime	DEMO_UTC:VINTERSOMMER
2023 03 25 23:00	-22
2023 03 26 00:00	-23
2023 03 26 01:00	0
2023 03 26 03:00	1
2023 03 26 04:00	2
2023 03 26 05:00	3
2023 03 26 06:00	4

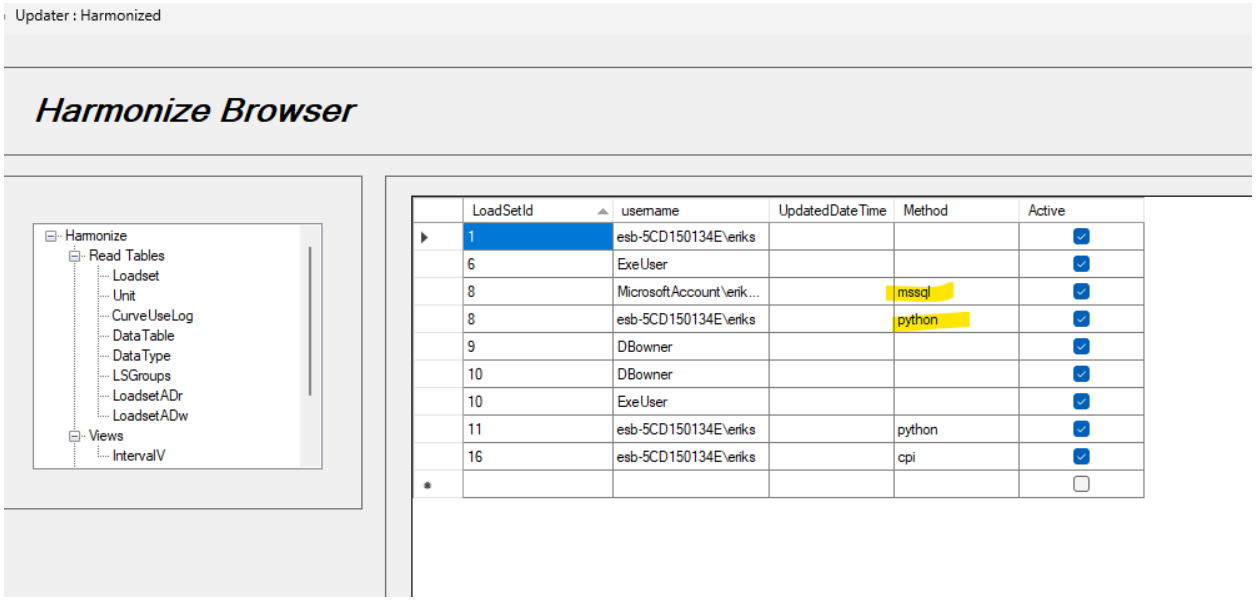
Error messages:

- **Cannot connect to server**, verify username and password, or the trusted connection, if possible, try with sql management studio for better error messages, or login with an admin user to see if password has expired.

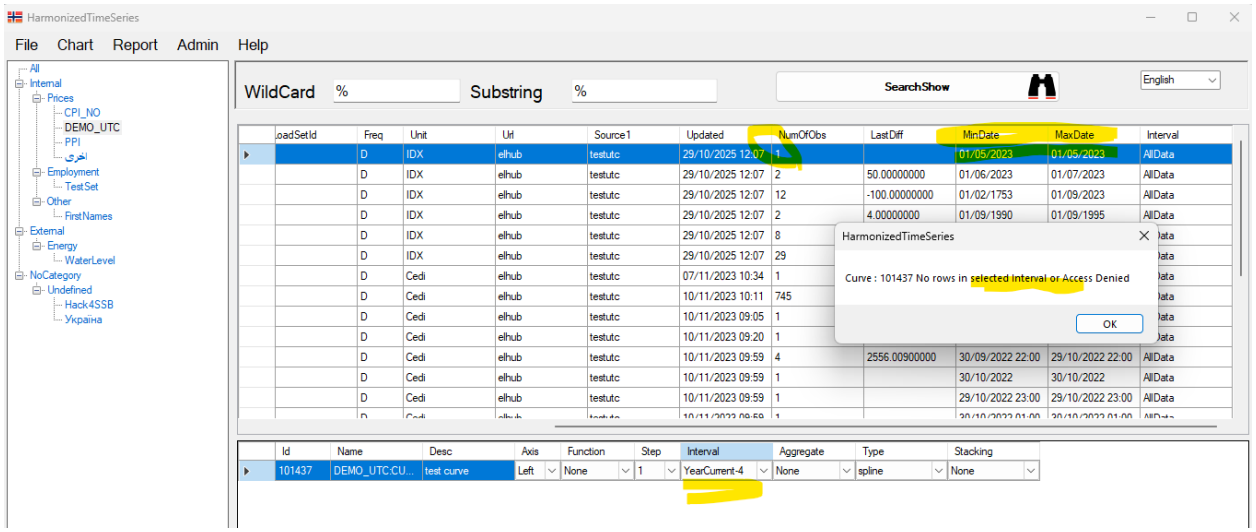
- When installing / reinstalling, it may be the installer is not able to remove previous installation, then settings, App – Uninstall Harmonize 'manually'

- Cannot update /"upsert" data into the database. This is problem related to database access. When running the application with sa (sysadmin) you do not necessarily have write access to the database. The access-all flag in the *loadset* table indicates whether all users have read access to the dataset. If/ When not all users should have access, the table called LoadsetADr, configure which users have access to which loadset = dataset.

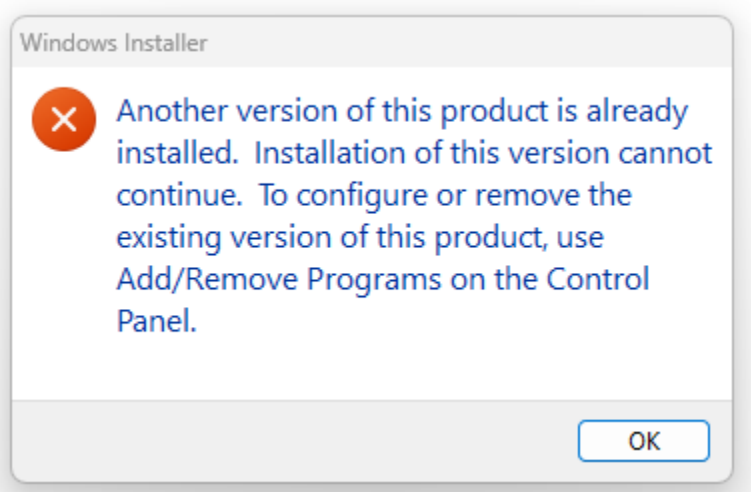
- The table LoadsetADw will indicate which users have write access to the datasets. Be aware that logging into the database with trusted connection using mssql management, may not be the same username as when logged in via trusted connection using Python. When using python your trusted user is typically the username printed with the *cmd* windows- command ***whoami*.** Users who need write access to the datasets must be defined in the table *LoadsetADw. To shortcut the installation and get started process, all users typically have write access to the database. The write access functionality is defined in the database function IsAccessw. The content of the table LoadsetADw shown below gives you the idea.*



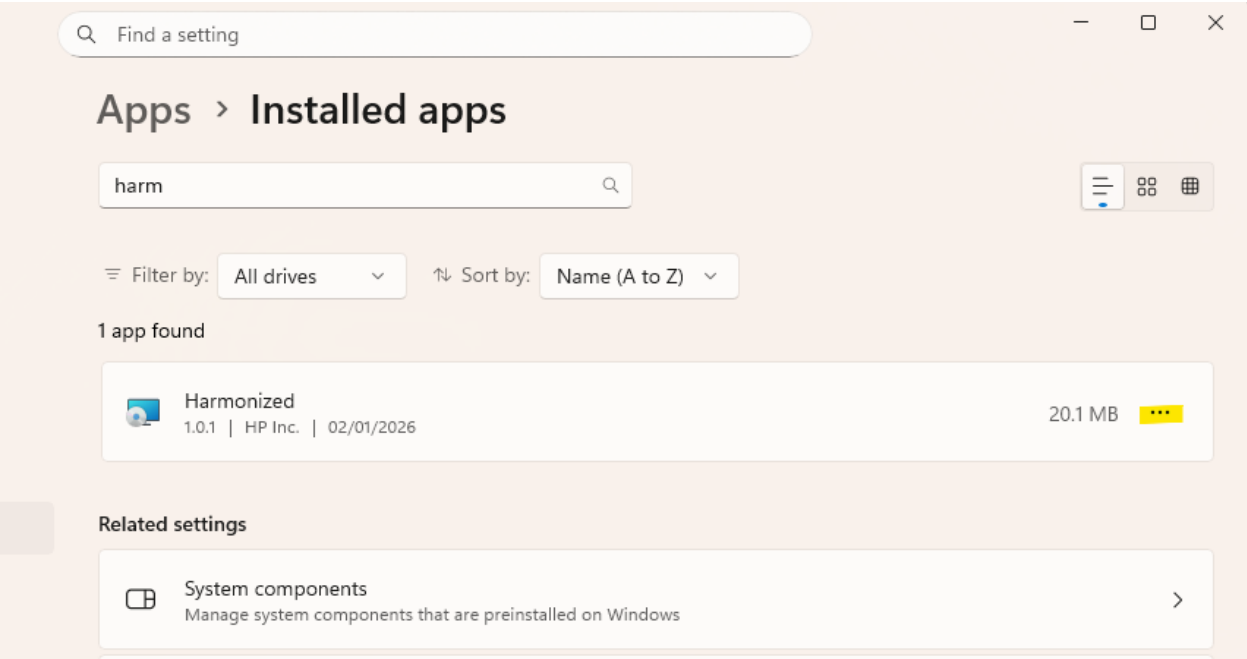
- No Access ***or no rows/data in selected intervals***. Make sure you have data in the selected interval. If your interval is the year 4 years ago, and there is no data, you will get this warning. Here you must change the interval, also be aware of the metadata indicating the start and last observation, also be aware charting a series with one only observation, the chart may look empty because it's only one point. This error may also occur if you have selected a common base year or a function that requires data in a interval, that does not exist or is empty.



- If setup is already installed, you will need to use windows add/remove applications and remove the Harmonize software. Then run the setup.exe again.



Windows + I, and Apps will take you to Installed Apps



Notes and Best Practices

- **MyData.js** is overwritten with each chart update.
- Logic and computation mainly occur **inside the database**.
- Use a trusted connection to sql if possible

- C# application Harmonize is lightweight:
    - Calls stored procedures in the MSSQL database with parameters
    - Updates `MyData.js` for charts, the same MyData.js used for all templates, enabling you to just simply change template chart-multichart-report from the browser.

- Use python or R to modify and update data, use stored procedures that control the access rules

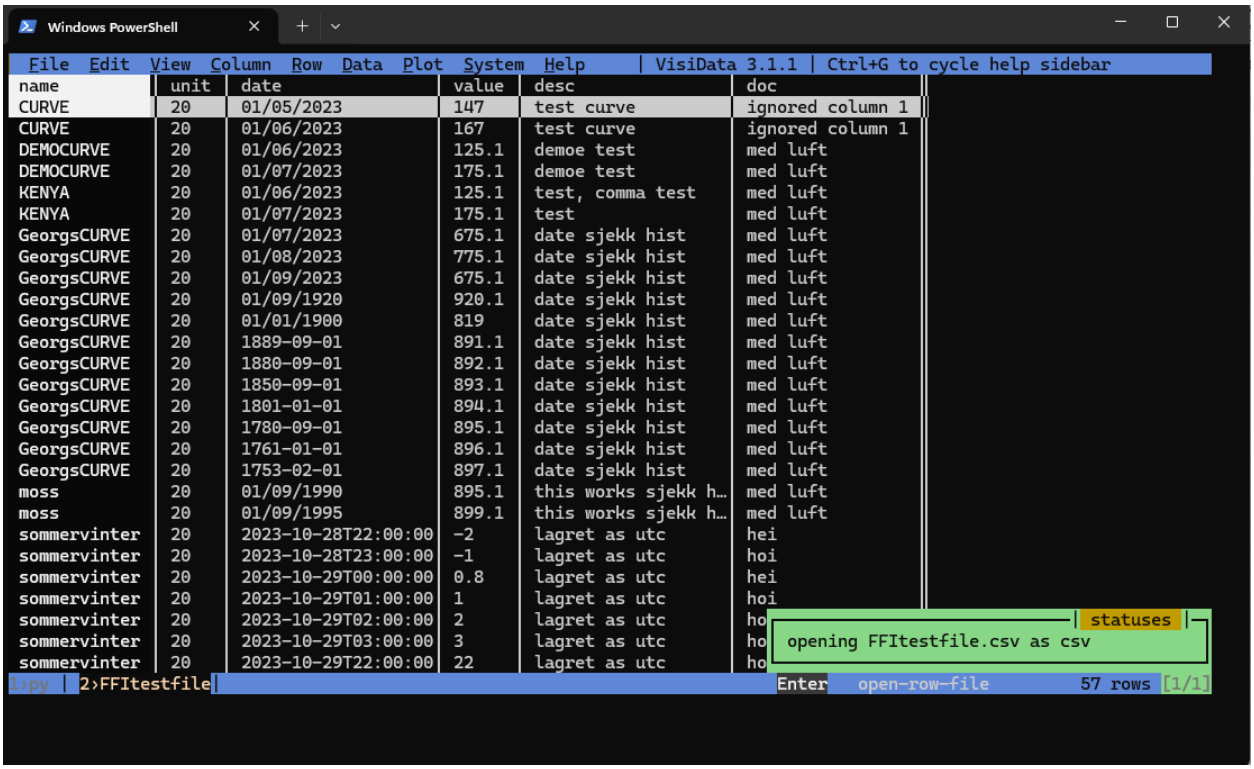
The application works without internet connection, but you must have access to a MSSql server to extract new data from a Harmonize database model.

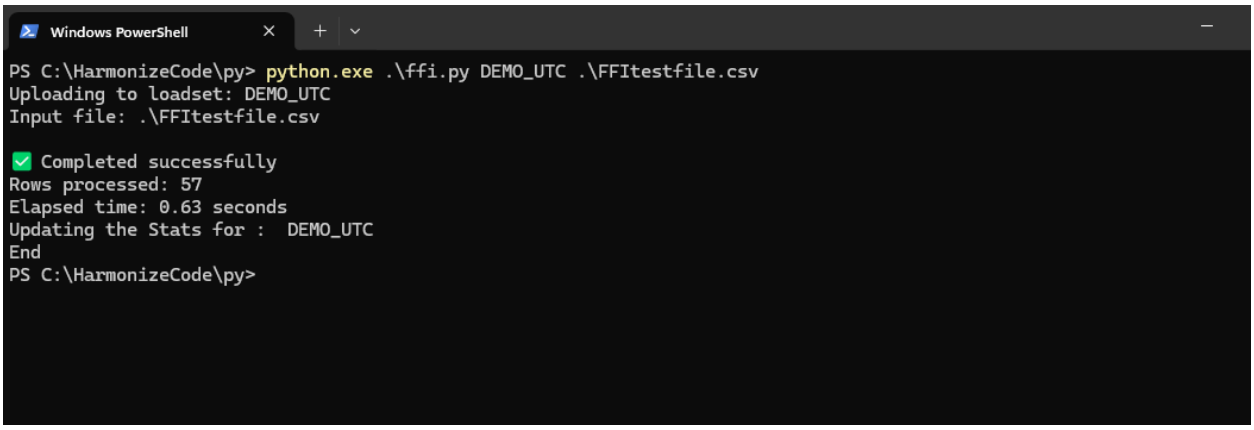
Python

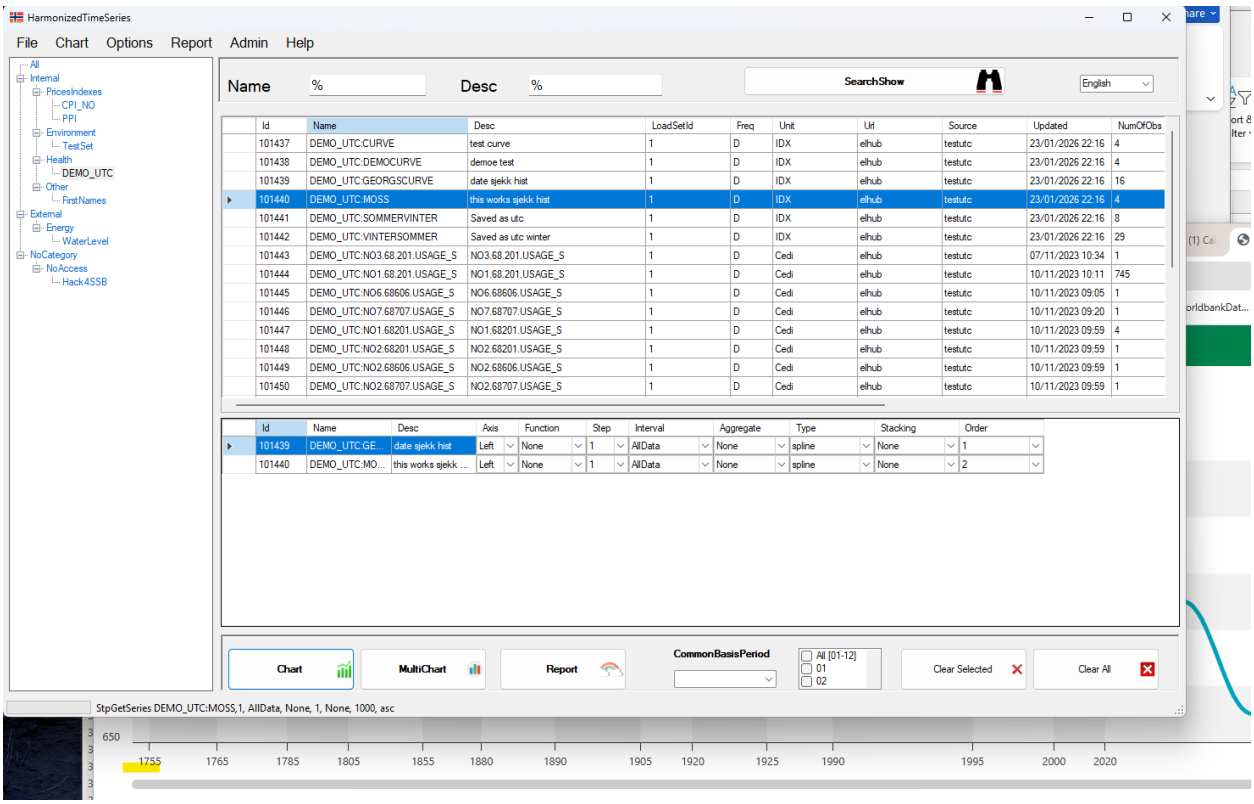
Import Update Upsert

A generic ffi.py flat file interface, comes with the system to provide generic upload of datafiles in a standard csv format, where comma is the separator. Commas my be included in text such as description or documentation but will need to be double quoted.

Typically, one dataset, will have same frequencies and date format, but there are no built-in restrictions for this. Data should be loaded in UTC time zone, where time zone conversions will apply.





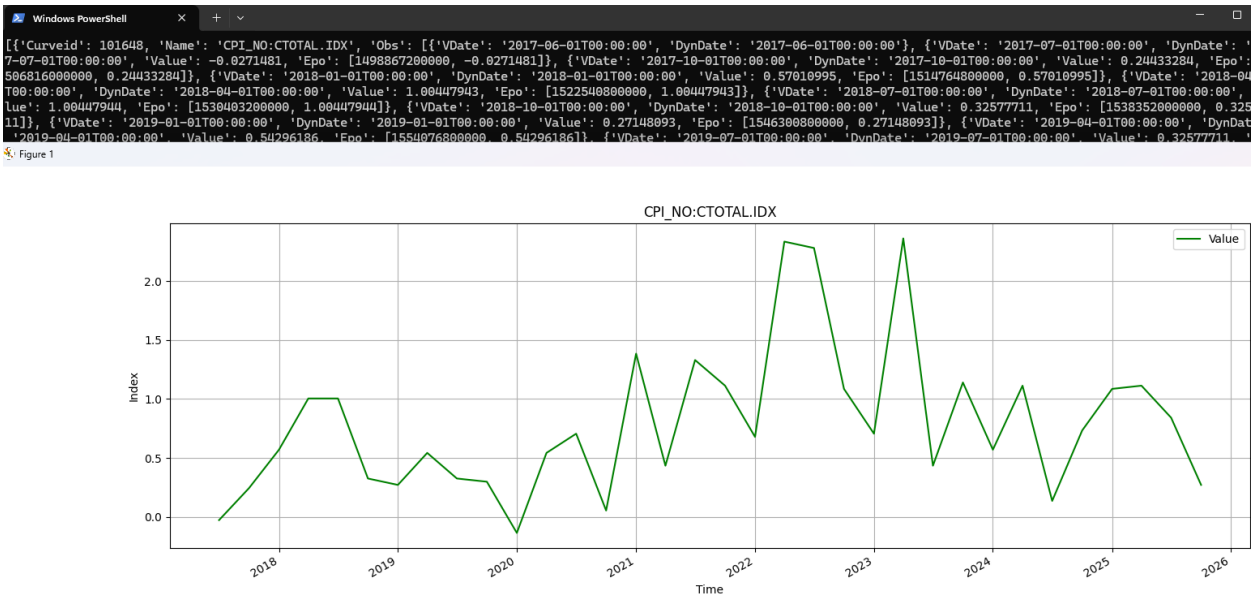


After the file has been loaded into Harmonize. The system is using the name of the dataset as prefix, making it possible to have more versions of the dataset in the system, and compare them.

Note timeseries from year 1735 will cope with the x-axis and timeline. Other systems of struggle with extreme history like this.

Export data using python.

Sample script to export data comes with the system. The procedure to get data is the exact same procedure as the Harmonize.exe application is using and will ensure same result, if used with the same parameters, such as interval function etc.




```python
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
#import matplotlib.dates
import json
#pip install matplotlib


#samples:
#StoredProc = "execute Harmonize..StpGetSeries 'CPI_NO:CTOTAL.IDX',2014,'Default',NULL,1,'json', 'NONE',1500, 'asc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'AllData',NULL,1,'json', 'NONE',1500, 'asc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'AllData',NULL,1,'json', 'NONE',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2',NULL,1,'json', 'NONE',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2',NULL,1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2',NULL,1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2','PCT(n)',1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2','[diff(n)]',1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"
#StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX',NULL,'YearsLast2','[diff(n)]',1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"
#baseyear and function
StoredProc = "execute StpGetSeries 'CPI_NO:CTOTAL.IDX','2022','YearsLast10','[diff(n)]',1,'json', 'AVG_QUARTER',1500, 'desc', 'NULL'"


#sjd use ur built in login trusted connection, can be in separete file or use existing connection.txt
conn = pyodbc.connect('Driver={SQL Server};'
                      'SERVER=localhost\\MSSQLSERVER2022;'
                      'Database=Harmonize;'
                      'Connection Timeout=5;'
                      'Integrated Security=True;')
```

Sample calls from python, using different parameters.

Utilities

To insert delete or update data, it's strongly recommended to use the Utility procedures in harmonize. This will ensure correct updating of automatic metadata, and use of existing write access mechanisms.

The procedures to *upsert* and delete data is available inside the database, as stored procedures. See also the python sample scripts to update the database correctly. In a timeseries database data observations must be stored with date or datetime, implying that storing information with a "date format" 2020M12 is not possible, but a proper date format like 2025-12-01 must be used. By storing the information in a correct manner, it then will be easier to compare information that may be stored in formats such as 2025, 2025M12 and 2025W52, making it impossible to compare without converting to date format.

- **DELETE_DATA**

```
exec UTILS_Delete_Loadset 'DELETE_DATA', 7, 'NameofDatasetToBeDeleted'
```

The DELETE_DATA option deletes all datapoints on all series on a dataset. The series names/ curvenames and the configuration of loadset will remain.

- **DELETE_SERIES**

```
exec UTILS_Delete_Loadset 'DELETE_SERIES ',7, 'NameDatasetToBeDeleted'
```

The DELETE_SERIES option deletes all datapoints on all series on a dataset as well as the names/ curvenames. However, the configuration of loadset will remain. This option is the best to use, when creation a *loadset*, but you find out you are not happy with the name of your series.

- **DELETE_ALL**

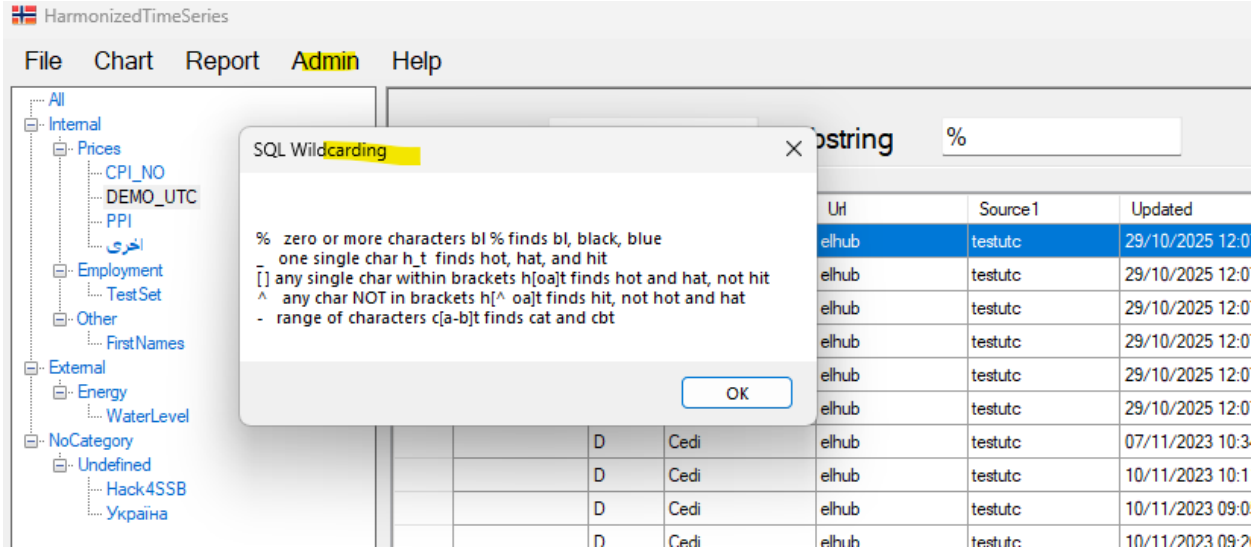
```
exec UTILS_Delete_Loadset 'DELETE_ALL ',7, 'NameDatasetToBeDeleted'
```

The DELETE_SERIES option deletes all datapoints on all series on a dataset as well as the names/ curvenames AND configuration of loadset. Be Careful. You need both the *Loadset_ID* and *LoadsetName* as argument when executing Util_Delete_loadset.

Recommended Workflow & Checklist

- Set up the **SQL Server database**
- Install Harmonize (**`Setup.exe`**)
- Configure database connection details in: *connection.txt*
- Run application and generate charts verify, Multichart.html, and the Report.html
- All *html* files can be improved or customized, feel free.
- Possible to add extra languages
- Use sample python script to create your own to import your data for shared analytics
- The database by default let all users have access to all datasets by **Access all** flag in the *loadset* table, (user access can be set by defining users and dataset, in the access tables LoadsetADr and LoadsetADw)
- Logging can be configured.
- If you do NOT have access to a dataset, the dataset will be listed in the tree, but no series will be listed when searching
- Be aware that wildcard and text search start searching from your position in the tree, if All is selected you will search all databases, if Prices group is selected you will search the databases in the Prices group.

- Search by wildcard for names and/or descriptions using standard sql **LIKE** syntax.

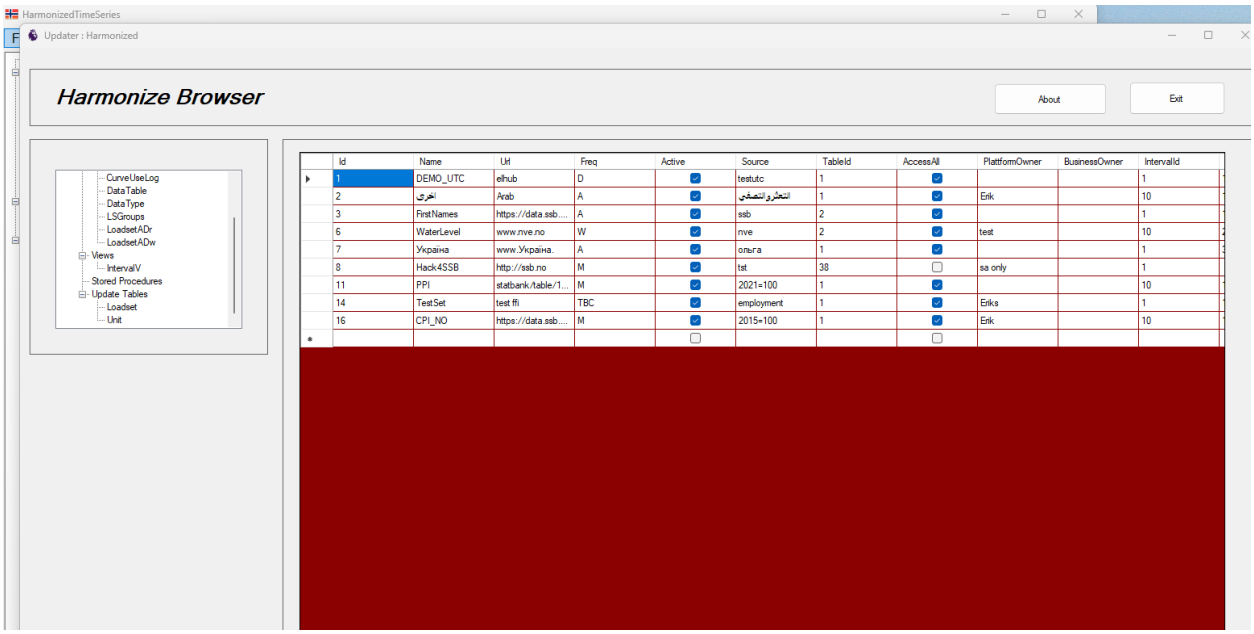


- Functions coming with the system are Mave(n), Diff(n) and Pct(n), it's obviously possible to create and add your own customized functions, if the functions are in the same style and use the same parameters, returning the same format. Additional Functions can be placed in the database and will be available in the drop down inside the application.

- Similar, additional intervals can also be defined in the database, and all intervals defined will be selectable in the application. Make sure to use a default interval that is user friendly for most people, this is not necessarily All data

Admin tool

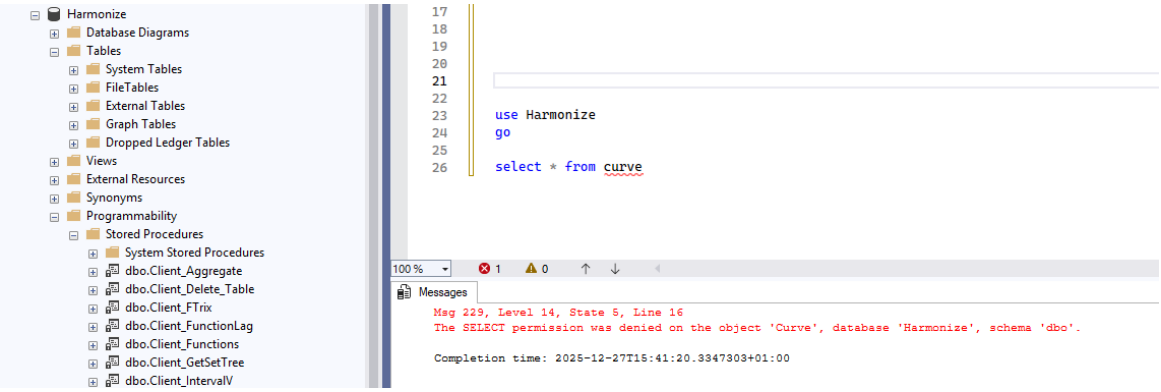
Admin is a browser for users to see some of the tables, and some of the tables can in theory be updated from here, but it's mainly a tool for viewing data such as units, access, intervals and more. The form should come up like this:



When you open the loadset table in Update mode – it will display in a red color mode. Depending on constraints on the various tables it is possible to do updates, but to update tables from sql management server will always be even better.

The advantage of the form is the use of sql procedures, making it possible for users to see data without select permissions.

Appendix – create database user script



By design users cannot make pure selections like this, data tables may be huge, therefore only correct queries are recommended.

Create user script on the following page, need to be an administrator sa to execute

```
USE master;
GO

IF EXISTS (SELECT 1 FROM sys.server_principals WHERE name = 'demo')
BEGIN
    DROP LOGIN demo;
END
GO

CREATE LOGIN demo
WITH PASSWORD = 'YourStrongPass***…',
     CHECK_POLICY = ON,
     CHECK_EXPIRATION = OFF,
     DEFAULT_DATABASE = master;
GO
USE Harmonize;
GO
-- IMPORTANT after restore
ALTER AUTHORIZATION ON DATABASE::Harmonize TO sa;
GO
DROP USER IF EXISTS demo;
GO
CREATE USER demo FOR LOGIN demo;
GO
GRANT CONNECT TO demo;
GRANT EXECUTE TO demo;
GO
USE master;
GO

ALTER LOGIN demo WITH DEFAULT_DATABASE = Harmonize;
GO
```