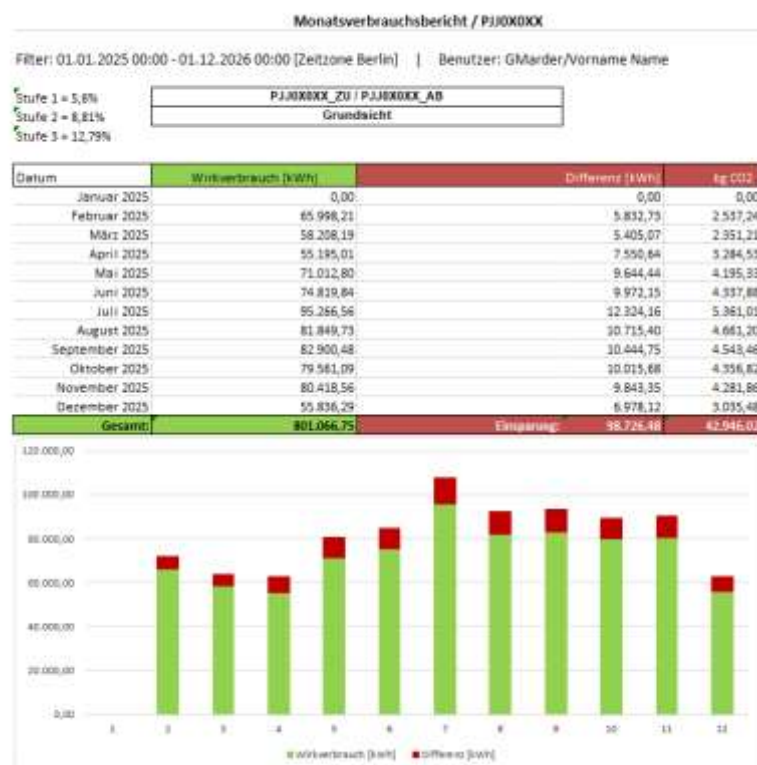


# eSaver-Ausw\_Erspar v1.02.3

## Bedienungsanleitung



## Allgemeine Hinweise

Um die einzelnen Stunden-Verbrauchs-Berichte pro Monat in die auszuführende Excel Datei zu kopieren, wurde dieses Tool erstellt.

Es soll helfen, die Übernahme der Daten in die Auszuwertende Datei zu erleichtern und Fehler beim Umsetzen zu vermeiden.

**Inhaltsverzeichnis**

<b>1. Aufbau der Konfigurationsdatei</b>	<b>4</b>
1.1. Besonderheiten der Konfigurationsdatei	4
1.1.1. Voraussetzungen für die Nutzung des Tools	4
<b>2. Beschreibung der Konfigurationsdatei</b>	<b>5</b>
2.1. Aufbau der Tabelle 1 [Export-VIDA Main]	5
2.2. Bedienung der Excel-Datei	6
2.2.1. Button zum Testen der Verbindung zu Python	6
2.2.2. Button zum Bearbeiten der VIDA-Daten	7
<b>3. Beschreibung der Xlwings Konfigurationsdatei</b>	<b>8</b>
3.1. Aufbau der Tabelle 2 [xlwings.conf]	8
3.2. Sonderfunktion von Xlwings	9
3.2.1. Sonderfunktion "Debug UDFs"	9
3.2.2. Sonderfunktion „Show Console“	9
3.2.3. Sonderfunktion "Use UDF Server"	10

## Bedienungsanleitung

### 1. Aufbau der Konfigurationsdatei

Als Konfigurationsdatei wurde eine Excel-Datei „eSaver\_Export\_VIDA\_Ausw\_Erspar\_SVB\_PJJ0X0XX.xlsm“ erstellt.

Kürzel im Namen:

- SVB: Stundenverbrauchsbericht
- PJJ0X0XX: Projektnummer

#### 1.1. Besonderheiten der Konfigurationsdatei

Die Excel-Datei enthält alle benötigten Informationen zu den Dateien und Verzeichnissen.

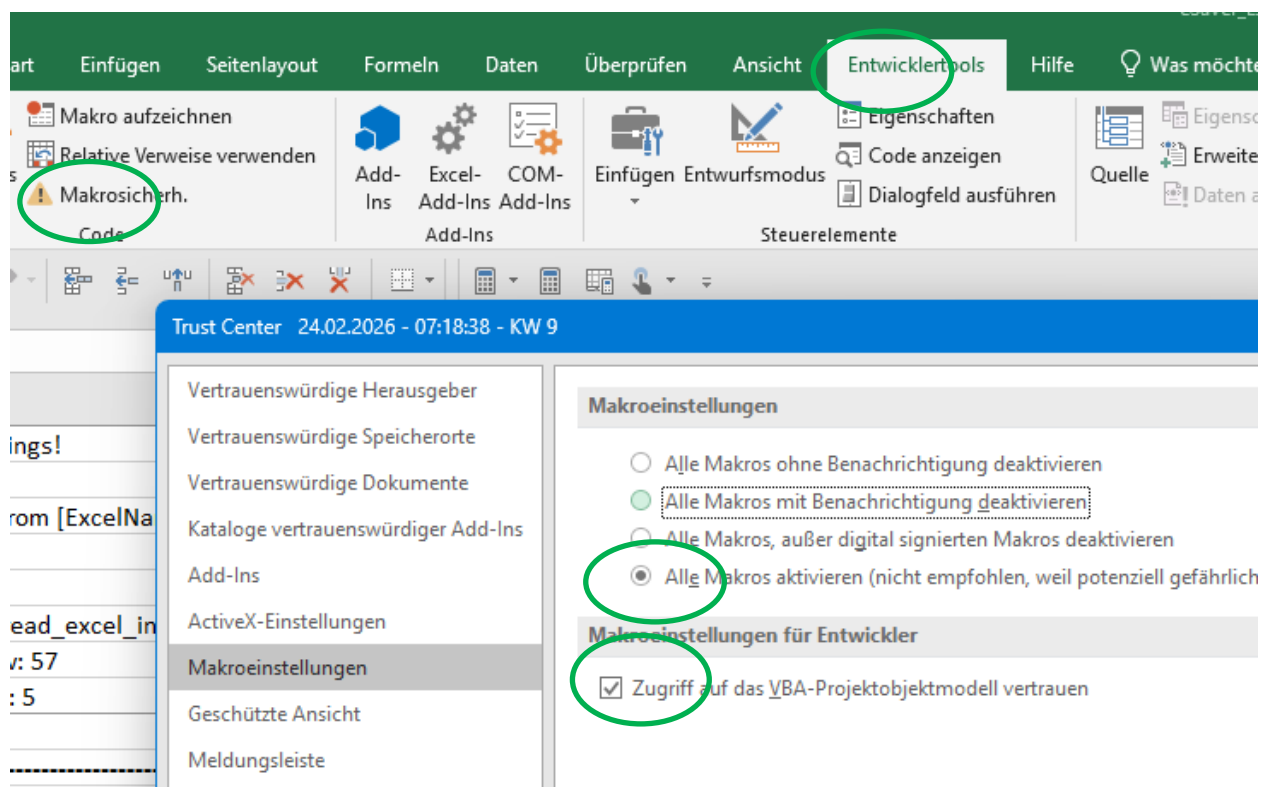
Die weitere Bedienung des Tools erfolgt ebenfalls über diese Excel-Datei.

Die Steuerung erfolgt über definierte Schaltflächen.

##### 1.1.1. Voraussetzungen für die Nutzung des Tools

Die Datei wurde für den Zugriff durch VBA freigegeben.

Dies ist erforderlich, damit über Python-Packages (xlwings) eine Zusammenarbeit zwischen Excel-Datei und Tool ermöglicht wird.



## Bedienungsanleitung

## 2. Beschreibung der Konfigurationsdatei

### 2.1. Aufbau der Tabelle 1 [Export-VIDA Main]

Tabelle: [Export-VIDA Main]

Bye xlwings!	Initial-Werte
Called from [ExcelName_py_xw_main_csv_to_excel_req:]	Test Main
call_2: read_excel_inits Fertig!	d:\20_PROJ_ABWICKL_eSaver_Ausw_Erspar\esaver_Ausw_Erspar_Py_v1.02.3.4_py31404\src
last_row: 57	Step 1: Datei Excel Initialisierung
last_col: 5	Step 2: gen Import Generierung
	Step 3: Datei Auswertung Importierung
eSaver Export VIDA Aus-Erspar INIT's	Name1-Name2
eSaver:	GE3 M / 1000 kVA
Projektnummer:	PJ0000XX
Datum	18.02.2026
Jahresübersicht	2025
Datum_Start	01.01.2025
Datum_End	31.12.2025
VIDA-Version	GridVis
Formatvorlage	*.csv
Export VIDA	Stundenverbrauch
Path_Export_Files	d:\20_PROJ_ABWICKL_eSaver_Ausw_Erspar\esaver_Ausw_Erspar_Py_v1.02.3.4_py31404\example\
Path_Import_Files	d:\20_PROJ_ABWICKL_eSaver_Ausw_Erspar\esaver_Ausw_Erspar_Py_v1.02.3.4_py31404\example\
Export_File_01_Jan	PJ0000XX Export_VIDA-Data_2025_0101_2025_0201_StdVerBer_GM.csv
Export_File_02_Feb	PJ0000XX Export_VIDA-Data_2025_0201_2025_0301_StdVerBer_GM.csv
Export_File_03_Mrz	PJ0000XX Export_VIDA-Data_2025_0301_2025_0401_StdVerBer_GM.csv
Export_File_04_Apr	PJ0000XX Export_VIDA-Data_2025_0401_2025_0501_StdVerBer_GM.csv
Export_File_05_Mai	PJ0000XX Export_VIDA-Data_2025_0501_2025_0601_StdVerBer_GM.csv
Export_File_06_Jun	PJ0000XX Export_VIDA-Data_2025_0601_2025_0701_StdVerBer_GM.csv
Export_File_07_Jul	PJ0000XX Export_VIDA-Data_2025_0701_2025_0801_StdVerBer_GM.csv
Export_File_08_Aug	PJ0000XX Export_VIDA-Data_2025_0801_2025_0901_StdVerBer_GM.csv
Export_File_09_Sep	PJ0000XX Export_VIDA-Data_2025_0901_2025_1001_StdVerBer_GM.csv
Export_File_10_Okt	PJ0000XX Export_VIDA-Data_2025_1001_2025_1101_StdVerBer_GM.csv
Export_File_11_Nov	PJ0000XX Export_VIDA-Data_2025_1101_2025_1201_StdVerBer_GM.csv
Export_File_12_Dez	PJ0000XX Export_VIDA-Data_2025_1201_2026_0101_StdVerBer_GM.csv
Import_File_01_Jan	PJ0000XX Export_VIDA-Data_2025_0101_2025_0201_StdVerBer_imp.xlsx
Import_File_02_Feb	PJ0000XX Export_VIDA-Data_2025_0201_2025_0301_StdVerBer_imp.xlsx
Import_File_03_Mrz	PJ0000XX Export_VIDA-Data_2025_0301_2025_0401_StdVerBer_imp.xlsx
Import_File_04_Apr	PJ0000XX Export_VIDA-Data_2025_0401_2025_0501_StdVerBer_imp.xlsx
Import_File_05_Mai	PJ0000XX Export_VIDA-Data_2025_0501_2025_0601_StdVerBer_imp.xlsx
Import_File_06_Jun	PJ0000XX Export_VIDA-Data_2025_0601_2025_0701_StdVerBer_imp.xlsx
Import_File_07_Jul	PJ0000XX Export_VIDA-Data_2025_0701_2025_0801_StdVerBer_imp.xlsx
Import_File_08_Aug	PJ0000XX Export_VIDA-Data_2025_0801_2025_0901_StdVerBer_imp.xlsx
Import_File_09_Sep	PJ0000XX Export_VIDA-Data_2025_0901_2025_1001_StdVerBer_imp.xlsx
Import_File_10_Okt	PJ0000XX Export_VIDA-Data_2025_1001_2025_1101_StdVerBer_imp.xlsx
Import_File_11_Nov	PJ0000XX Export_VIDA-Data_2025_1101_2025_1201_StdVerBer_imp.xlsx
Import_File_12_Dez	PJ0000XX Export_VIDA-Data_2025_1201_2026_0101_StdVerBer_imp.xlsx
PJ0000XX_Ersparnis_File	PJ0000XX_Name1-Name2 Ersparnis 2025_01-12 - Stundenbasiert GMOB.xlsx

In dieser Tabelle werden die Parameter mit den Einstellwerten hinterlegt.

## Bedienungsanleitung

---

### 2.2. Bedienung der Excel-Datei

Zur Vereinfachung der Konvertierung der Dateien sowie der Übernahme der Werte in die Jahresauswertungsdatei wurden entsprechende Bedienbuttons integriert.

#### 2.2.1. Button zum Testen der Verbindung zu Python



##### ❖ Call\_Py\_xw\_main\_req()

Damit wird die Funktion "xw\_main\_req" in Python aufgerufen, und eine Nachricht in die Zeile A1 geschrieben

-> Der Text in Zelle A1 wird pro betätigen des Button abwechselnd mit ‚Hello xlwings‘ / ‚Bye xlwings‘ geschrieben, um eine Rückmeldung von Python zu erkennen

-> Der Text in Zelle A3 wird ‚Called from [ExcelName\_py.xw\_main\_req:]‘ eingetragen  
In dieser Zelle werden auch die Rückmeldungen von den Bearbeitungsschritten eingetragen.

## Bedienungsanleitung

---

### 2.2.2. Button zum Bearbeiten der VIDA-Daten



❖ **Schritt 1: Datei Excel Initialisierung**

Damit wird die Funktion zur Initialisierung der Excel-Importeinträge in Python aufgerufen. Die Dateinamen werden aus den Spalten „Export-Files\_MM\_mmm“ gelesen und die Erweiterung in „\_Imp.xlsx“ geändert. Diese neuen Dateinamen werden anschließend in die Spalten „Import\_Files\_MM\_mmm“ übernommen.

❖ **Schritt 2: Datei Import Generierung**

Die Exportdateien werden eingelesen, die enthaltenen Werte überprüft und auf eine einheitliche Struktur konvertiert. Anschließend werden die entsprechenden Importdateien erzeugt.

❖ **Schritt 3: Datei Auswertung Importierung**

Die generierten Importdateien werden eingelesen und die enthaltenen Daten überprüft. Anhand des Datums wird die passende Tabelle in der Auswertungsdatei geöffnet und die Werte werden übertragen.

### 3.1. Aufbau der Tabelle 2 [xlwings.conf]



### 3.2. Sonderfunktion von Xlwings

#### 3.2.1. Sonderfunktion “Debug UDFs”

Debug UDFs	TRUE
------------	------

➤ **TRUE**

Die Debug unter VBA wird unterstützt

➤ **FALSE**          Standard

Die Debug unter VBA wird nicht unterstützt

#### 3.2.2. Sonderfunktion „Show Console“

Show Console	TRUE
--------------	------

➤ **TRUE**

Die Windows-Konsole mit den Python „print()“-Ausgaben“ wird angezeigt.

Info:

Die Abläufe der Funktionen sind langsamer,  
durch die optische Rückmeldung können Fehler leichter diagnostiziert werden.

➤ **FALSE**          Standard

Die Windows Konsole mit den Python „print()“-Ausgaben“ wird nicht angezeigt.

Info:

Der Ablauf der Funktionen ist dadurch geringfügig schneller,  
bei Fehler bekommt man aber keine optische Rückmeldung angezeigt.

## Bedienungsanleitung

### 3.2.3. Sonderfunktion "Use UDF Server"

Use UDF Server	FALSE
----------------	-------

➤ **TRUE**

Es wird eine Abfrage aktiviert, die den Start des Debuggers von VSCode erkennt und eine Verbindung zum Debug-Mode von VSCode herstellt.

Nun kann unter VSCode die Master-Datei im Debugging-Mode gestartet werden.

Zur Information wurden Hinweise im Programm hinterlegt.

```
eSaver_Export_VIDA_Ausw_Erspar_SVB.py > ...
Zelle Ausführen | Oben laufen | Zelle debuggen
# %% === xw_serve
def xw_serve():
    """Starten von Server für debug

    Danach könne die Werte in Excel gestartet werden,
    um die Scripte zu debuggen

    Note:
        unter EXCEL muss 'Use UDF Server' auf TRUE stehen
        ohne xw_server() muss aber zwingend FALSE eingetragen sein.

    """
    # ref: https://docs.xlwings.org/en/stable/debugging.html

    xw.serve() # readme Einstellungen in Excel [xlwings.conf] beachten

    return
```

You, letzte Woche • Neu Janitza mit Umrechnung und eCon in \_import-...

➤ **FALSE**      Standard

Es wird keine Abfrage an ein laufendes Python Programm gestartet.  
Ein debuggen unter VSCode funktioniert nicht,  
es kann keine Verbindung aufgebaut werden.  
Xlwings starten im standard-Mode eine eigene Instanz von Python,  
die dadurch keine Verbindung zu VSCode aufbauen kann.