Synkronizer 11 User Manual

Table of Contents

Part I

1 Welcome

Dear Synkronizer users

Thanks for choosing Synkronizer 11. Synkronizer 11 is the most powerful Excel comparison program in the market. Synkronizer 11 is the only program worldwide that can compare and update regular Excel files and Excel databases.

Synkronizer 11 allows you to

- compare Excel files,
- update / transfer of differences,
- highlight the differences in the underlying worksheets,
- create a difference report,
- merge different Excel worksheets or databases,
- find differences / duplicates,
- automate processes by using comparison projects,
- create syntax for command line and Visual Basic (VBA)

This manual introduces you to the Synkronizer 11 environment and explains the essential functions.

Of course Synkronizer 11 may not be perfect. We are always interested in learning from our users, so your feedback is always welcome. For any suggestions or questions (e.g. menu items, settings, forms, or about this manual) feel free to contact us directly.

Sincerely,

Thomas Strübi & Jurgen Volkerink
XL Consulting GmbH

1.1 What's new?

Although you might be already familiar with the previous vervions, Synkronizer 11 contains some new features that we would like to present:

- Synkronizer 11 is a COM add-in.

- The program works with the following versions of Excel:
  Excel 365 / 32 -bit and 64 -bit
  Excel 2016 / 32 -bit and 64 -bit
  Excel 2013 / 32 -bit and 64 -bit
  Excel 2010 / 32 -bit and 64 -bit

- The following operating systems are supported:
  Windows 10 / 32 -bit and 64 -bit
  Windows 8 / 32 -bit and 64 -bit
  Windows 7 / 32- bit and 64- bit
  Windows Vista / 32- bit and 64- bit
  Windows XP / 32- bit and 64- bit
  Windows Server 2012
  Windows Server 2008
  Windows Server 2016

- The program has a new user interface.

- Comments can be compared.

- Excel names can be compared.

- There are two new categories of differences: "Entered Values & Formulas" and "Calculated Values". These help to identify cell differences more accurately. The category "Entered values & formulas" will report different cell values or formulas. The "Calculated Values" type will report a difference in case identical formulas will have different.

- New filter functions.

- Invocation via command line. With the Developer Edition, you can control Synkronizer via the command line. The Developer edition will earliest be shipped sometime after summer 2014.

1.2 Editions

There are three different editions of Synkronizer 11 available. The functionality of these editions is described in the following table:

	Professional Edition	**Developer Edition**	**Trial Edition**
Compare all cells of spreadsheets	•	•	
Compare limited range (A1:Z100) of spreadsheets			•

Compare cell values	•	•	•
Compare cell formulas	•	•	•
Compare cell comments	•	•	•
Compare text formatting	•	•	•
Compare cell formatting	•	•	•
Compare Excel names	•	•	•
Compare tables in database mode	•	•	•
Show differences in results form	•	•	•
Highlight differences	•	•	•
Create a difference report	•	•	•
Update tables	•	•	•
Show/Hide differences	•	•	•
Create projects	•	•	•
Use filters	•	•	•
Create powerful comparing scripts		•	
Command line option		•	
VBA extension to create automated scripts		•	
Price	EUR 89.--	EUR 199.--	Free

1.3 Supported Excel versions & Operating Systems

Synkronizer 11 has only few standard requirements and can be installed on local PCs as well as on networks.

Supported Excel Versions

- Excel 365 / 32-bit and 64-bit
- Excel 2019 / 32-bit and 64-bit
- Excel 2016 / 32-bit and 64-bit
- Excel 2013 / 32-bit and 64-bit
- Excel 2010 / 32-bit and 64-bit

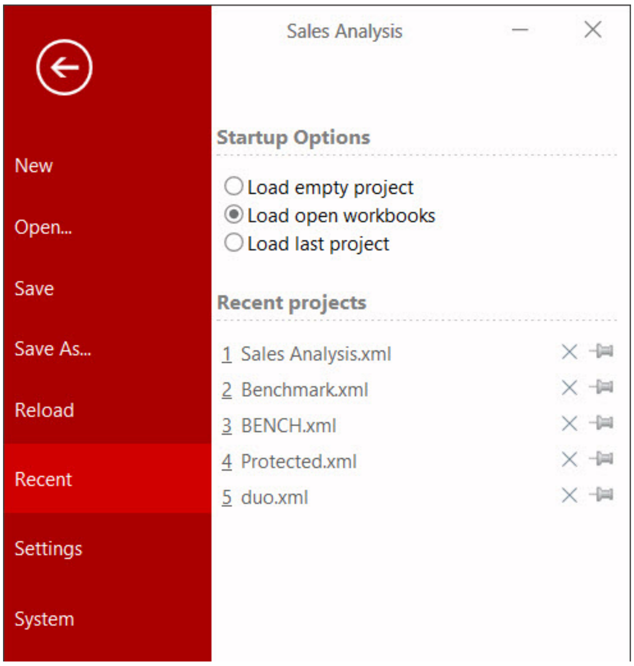
- **Supported Operating Systems**
- Windows 10 / 32-bit and 64-bit
- Windows 8 / 32-bit and 64-bit
- Windows 7 / 32-bit and 64-bit
- Windows Vista / 32-bit and 64-bit
- Windows XP
- Windows Server 2012
- Windows Server 2008
- Windows Server 2016

- Mac OS is not supported
  To run Synkronizer on a Mac you need to install a virtual Windows/Office on the mac.

Part II

2 Project

In this tab, you can manage projects, change project settings and find information about your Synkronizer edition.



Startup Options

○ Load empty project
● Load open workbooks
○ Load last project

Recent projects

1 Sales Analysis.xml
2 Benchmark.xml
3 BENCH.xml
4 Protected.xml
5 duo.xml

All comparison processes can be saved in projects. As a great benefit thereof, you don't need to select the files, worksheets, and change the comparison settings again each time. In this menu, the following commands are available to manage the projects:

New

The form is emptied, all settings are reset.

Open

This command opens an existing project.

Save

The current comparison settings (files, worksheets, format and filter settings) are saved as a project. Any passwords are not saved in the project file.

Save as ...

The selected project is saved under a new name.

Reload

The project is reloaded, all previous and unsaved changes are lost.

Startup Options

Here you can specify whether you want to start Synkronizer with an empty project, the last opened project or the opened files.

Recent / Recent projects

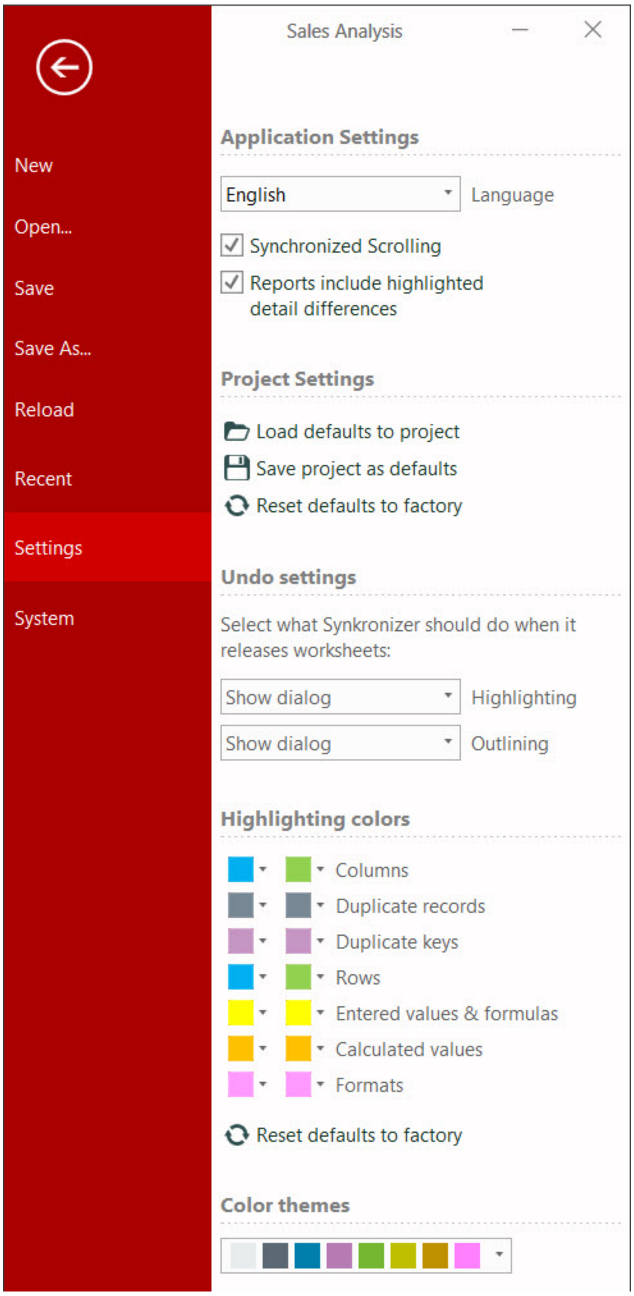
List of recently used projects. The project is reloaded, all previous settings are lost. Most project commands are also available in the Select tab.

Define general settings

Get/set information about your license

2.1 Settings

In this menu, general settings can be changed.



Application Settings

English ▾ Language

☑ Synchronized Scrolling

☑ Reports include highlighted detail differences

Project Settings

Load defaults to project

Save project as defaults

Reset defaults to factory

Undo settings

Select what Synkronizer should do when it releases worksheets:

Show dialog ▾ Highlighting

Show dialog ▾ Outlining

Highlighting colors

🟦 ▾ 🟩 ▾ Columns

⬛ ▾ ⬛ ▾ Duplicate records

🟪 ▾ 🟪 ▾ Duplicate keys

🟦 ▾ 🟩 ▾ Rows

🟨 ▾ 🟨 ▾ Entered values & formulas

🟧 ▾ 🟧 ▾ Calculated values

🟪 ▾ 🟪 ▾ Formats

Reset defaults to factory

Color themes

Application Settings

In this section you can define the general settings of Synkronizer.

- **Language**
  In this field, the language can be determined.
  If your language is not available, we will give you a free Synkronizer version (Developer Version) if you help us translate Synkronizer material into your mother-tongue. Just contact us for your free copy. Prior to starting your translation, please contact us for detailed instructions.

- **Reports include highlighted detail differences**
  If this option is activated, the detail information of text differences are displayed in color in the difference report. The disadvantage is that the creation of the difference report will be slower. If you disable this option, the difference report will be created faster.

- **Synchronized Scrolling**
  With this option you can scroll "synchronized" through the files. Synkronizer ensures that always the corresponding differences of the other file are displayed.

Project Settings

Here you can save, reload or reset your personal project settings. The following options are available:

- **Load defaults to project**
  Your preferred personal preferences are loaded (see "Save project as default")

- **Save project as default**
  With this button you can save your preferred settings. All settings regarding contents, formats, filters, report, highlighting and outline are saved.

- **Reset defaults to factory**
  All settings of the Select tab will be reset to the factory settings.

Undo settings

A strength of Synkronizer is its ability to highlighted differences and hide unwanted ones with the outlining feature. The downside of this option is that these changes will possibly remain in the Excel worksheets.

In this area you can choose what should happen with the highlighted and outlined changes. The following options are available:

- **Show dialog**
  When closing the Excel file, a dialog box is displayed allowing you to select whether the changes should be reset or not.

- **Never undo**
  The Synkronizer changes will not be reset.

- **Always undo**
  The Synkronizer changes are reset. Your Excel files will therefore be in the same state as prior to the comparison (except the transferred differences).

Color themes

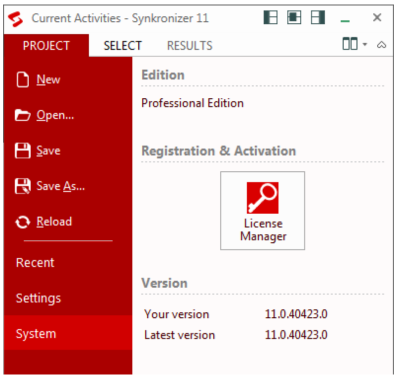
If you don't like the colors Synkronizer uses to highlight the differences, you have the option to define your own. The colors can be reset to the original "Synkronizer" colors with the button "Reset defaults to factory".

Do not use color

If certain difference types should not be highlighted, then simply select the desired difference type and click "No Color".

2.2 System

In this menu you will find some information about your Synkronizer license.



Professional Edition

Your version 11.0.40423.0
Latest version 11.0.40423.0

Edition

In this part you see the information about your edition

Registration & Activation

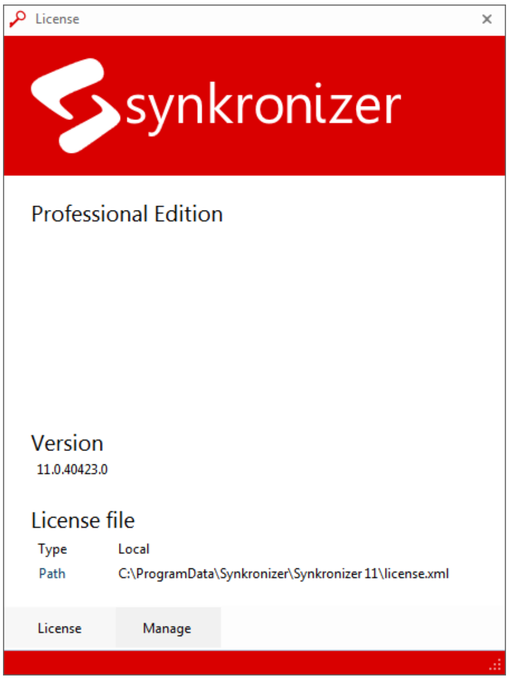
By clicking on the "License Manager" button, the license manager is started. In the License Manager you'll find more information about the installed license. You can also register, activate and deactivate the software. Furthermore you can administrate the nodes (network users) if you have a network license.

Version

In this part you can check if you are working with the latest build. If this isn't the case a download button appears with which you can download the latest build.

2.2.1 License Manager

In the license manager you can see information about your license .



Professional Edition

Version
11.0.40423.0

License file

Type Local

Path C:\ProgramData\Synkronizer\Synkronizer 11\license.xml

The following information is visible/can be set within this form:

- detailed information about your license
- register/activate license
- deactivate license
- define nodes/network users (applies only to network license)

2.3 FileFormat Project

The project file is a simple XML text file that can be edited with any standard text editor such as notepad.

When manually editing XML files, be aware that the names of tags are case-sensitive. Misspelling will cause failures.

Tip:
Prepare your project using the Synkronizer tab, save it, then edit it to suit your needs.

This is the layout of the file for a project where sheets are auto matched, without any options

set for any of the pairs.
You'll notice that there are no individual pairs listed inside the <Pairs> tag..

```
<?xml version="1.0" standalone="yes"?>
<Synkronizer>
   <Project ID="1">
      <Settings>
         <CompareType>0</CompareType>
         <Formats>0</Formats>
         <Filters>0</Filters>
         <FilterTolerance>0</FilterTolerance>
         <FilterPattern/>
         <ReportType>0</ReportType>
         <HighlightType>1</HighlightType>
         <ShowHide>0</ShowHide>
      </Settings>
      <Files>
         <Filename0>C:\Folder\SubFolder\File 1.xls</Filename0>
         <Filename1>C:\Folder\SubFolder\File 2.xls</Filename1>
      </Files>
      <Pairs>
         <MatchType>1</MatchType>
         <MatchInclude>2</MatchInclude>
      </Pairs>
   </Project>
</Synkronizer>
```

For a project with pairs of individual settings the file will look like this:
(MatchType will be set to 0 as soon as you select a pair option such as RangeAddr or DBKeys.)

The pair tags must be sequentially numbered via the ID attribute.

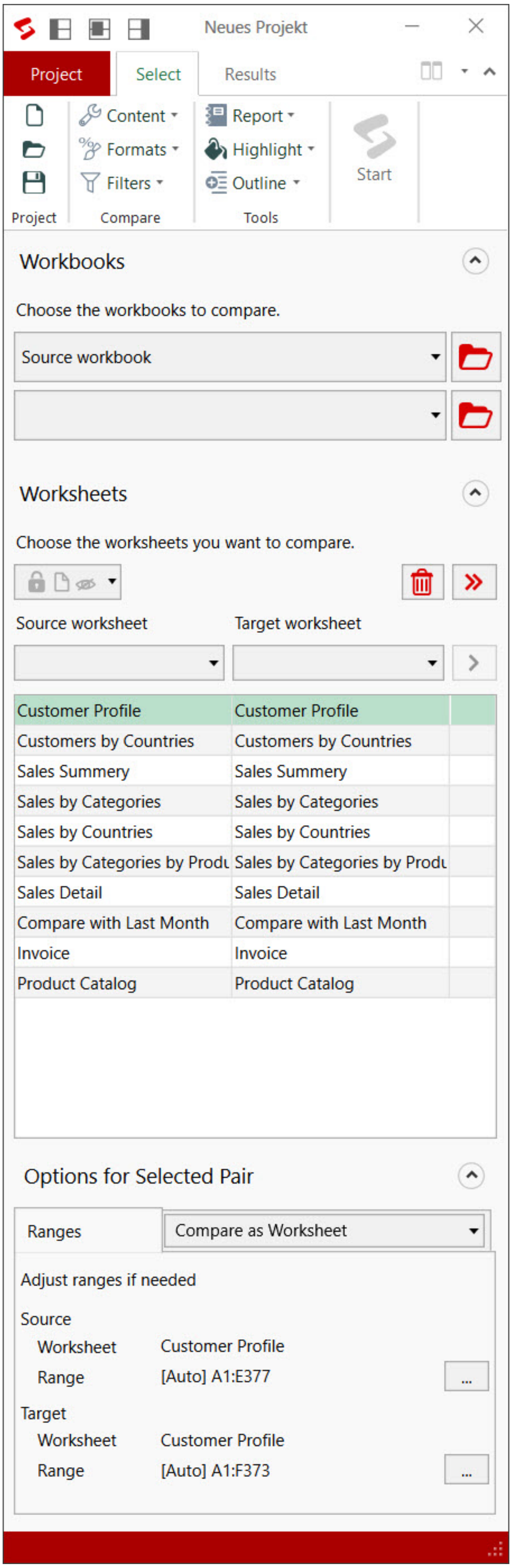
```
<?xml version="1.0" standalone="yes"?>
<Synkronizer>
   <Project ID="1">
      <Settings>
         <CompareType>0</CompareType>
         <Formats>0</Formats>
         <Filters>0</Filters>
         <FilterTolerance>0</FilterTolerance>
         <FilterPattern/>
         <ReportType>0</ReportType>
         <HighlightType>1</HighlightType>
         <ShowHide>0</ShowHide>
      </Settings>
      <Files>
         <Filename0>C:\Folder\SubFolder\File 1.xls</Filename0>
         <Filename1>C:\Folder\SubFolder\File 2.xls</Filename1>
      </Files>
      <Pairs>
```

```
         <MatchInclude>3</MatchInclude>
         <MatchType>0</MatchType>
         <PairCount>3</PairCount>
         <Pair ID="1">
            <SheetName0>Addresses</SheetName0>
            <SheetName1>Addresses</SheetName1>
         </Pair>
         <Pair ID="2">
            <SheetName0>Controlling</SheetName0>
            <SheetName1>Controlling</SheetName1>
            <RangeAddr0>$A$4:$AC$75</RangeAddr0>
            <RangeAddr1>$A$4:$AC$75</RangeAddr1>
            <DBRow>5</DBRow>
            <DBKeys>1</DBKeys>
            <DBOptionsMask>14</DBOptionsMask>
         </Pair>
         <Pair ID="3">
            <SheetName0>Budget</SheetName0>
            <SheetName1>Budget</SheetName1>
         </Pair>
      </Pairs>
   </Project>
</Synkronizer>
```

Part III

3 Select

When Synkronizer is started the main form or Select tab is displayed. The form is used to select entire projects or two Excel files that are to be compared, furthermore, virtually all comparison options, filters and actions can be defined in this form.



Workbooks

Choose the workbooks to compare.

Source workbook

Worksheets

Choose the worksheets you want to compare.

Source worksheet	Target worksheet
Customer Profile	Customer Profile
Customers by Countries	Customers by Countries
Sales Summery	Sales Summery
Sales by Categories	Sales by Categories
Sales by Countries	Sales by Countries
Sales by Categories by Produ	Sales by Categories by Produ
Sales Detail	Sales Detail
Compare with Last Month	Compare with Last Month
Invoice	Invoice
Product Catalog	Product Catalog

Options for Selected Pair

Ranges: Compare as Worksheet

Adjust ranges if needed

Source
Worksheet: Customer Profile
Range: [Auto] A1:E377

Target
Worksheet: Customer Profile
Range: [Auto] A1:F373

Ribbon

In this section are projects, comparison, tool and action options available. You can open, save or reset projects, define comparison options like contents, formats, filters, define the tools settings like report, highlight, outline and start the comparison process.

Select workbooks

Open and select the workbooks to be compared.

Select worksheets

Select the worksheets to be compared. The following options are available:
- define the worksheet types (all, protected, hidden)
- select and delete worksheets
- manually select worksheets

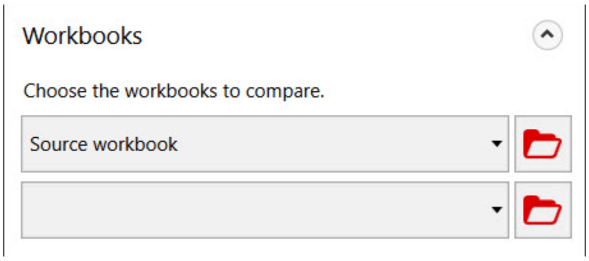
Options for worksheet pairs

Define worksheet settings like
- define ranges
- compare worksheets as database
- link worksheets 1 on 1
- compare as normal worksheet

The individual settings are described in the following chapters.

Note
All empty workbooks and all workbooks in protected view will be closed when the form starts.

3.1 Workbooks



Choose the workbooks to compare.

In this section you can specify which files are to be compared, in addition there are two combo boxes. The first file is entered in the upper box, and the second file in the lower box.

The Excel files can also be selected with the "Open File" symbols. Alternatively, the files can be selected using the combo boxes. If you click this combo box, a list of recently opened Excel files will appear.

Projects

If you have saved the files in a project, the files can also be opened using the menu command Project » Open. For details, refer to the Projects section.

Supported file types

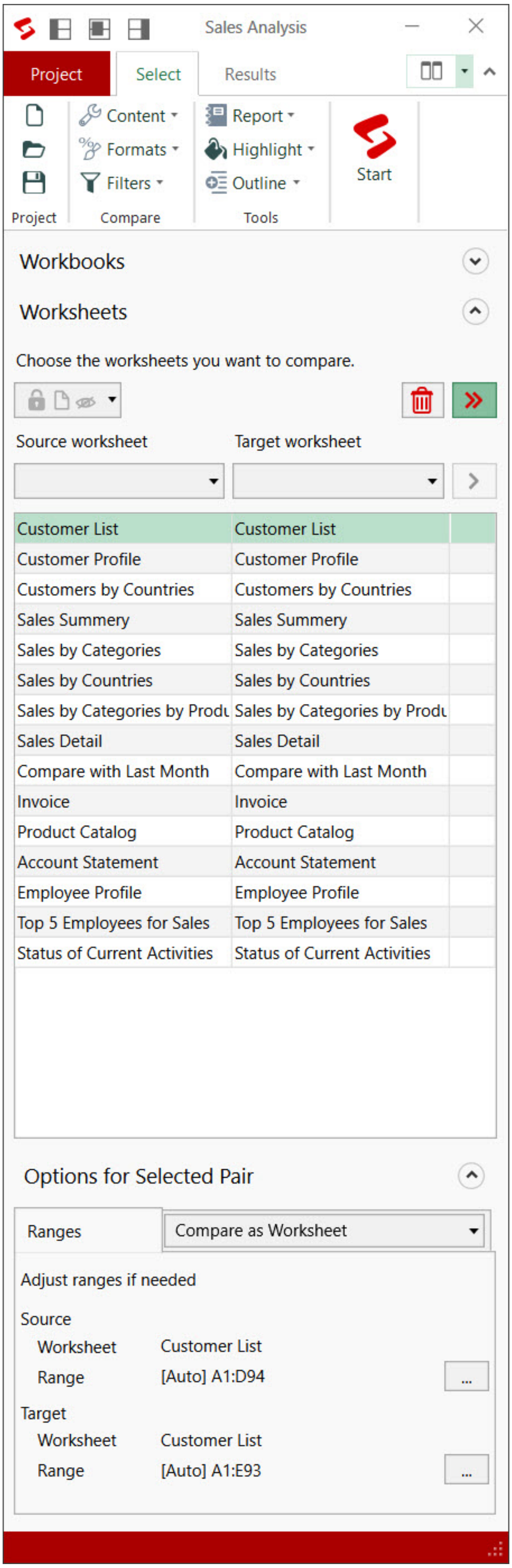
Synkronizer can open all files that are supported by Microsoft Excel.

Swap Workbooks

It is always possible to swap the workbooks at a later time. For doing this, simply click on the swap button in the "workbooks" area.

3.2 Worksheets

This section displays the worksheets are displayed which are to be compared. Moreover, in this section worksheet-specific settings such as ranges, database or comparison options are defined.



Workbooks

Worksheets

Choose the worksheets you want to compare.

Source worksheet	Target worksheet
Customer List	Customer List
Customer Profile	Customer Profile
Customers by Countries	Customers by Countries
Sales Summery	Sales Summery
Sales by Categories	Sales by Categories
Sales by Countries	Sales by Countries
Sales by Categories by Produ	Sales by Categories by Produ
Sales Detail	Sales Detail
Compare with Last Month	Compare with Last Month
Invoice	Invoice
Product Catalog	Product Catalog
Account Statement	Account Statement
Employee Profile	Employee Profile
Top 5 Employees for Sales	Top 5 Employees for Sales
Status of Current Activities	Status of Current Activities

Options for Selected Pair

Ranges: Compare as Worksheet

Adjust ranges if needed

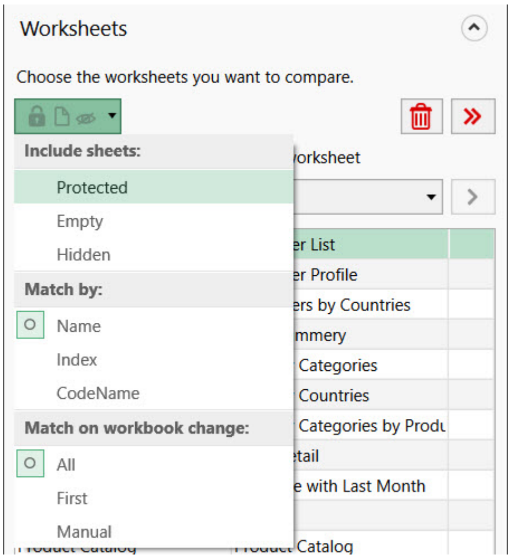
Source
Worksheet Customer List
Range [Auto] A1:D94

Target
Worksheet Customer List
Range [Auto] A1:E93

3.2.1 Settings

Automatically pair worksheets

When you select two files, Synkronizer automatically tries to pair the worksheets. The pairing settings of the worksheets are set in the following drop-down element:



The following options can be changed in this item:

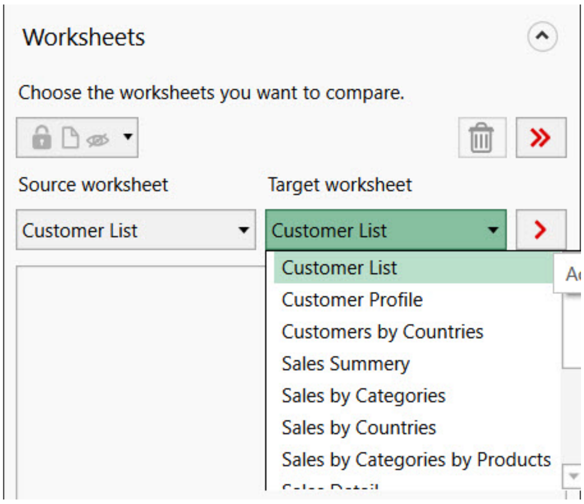
- **Include sheets**
  These types of worksheets are also listed.

- **Match by**
  Depending on the setting, the worksheets are matched by their names, index (position within the file) or codename (internal worksheet name).

- **Match on workbook change**
  Here you can specify how the worksheets are paired. If you select "all", all worksheets with the same name are transferred to the worksheets list, by selecting "first" only the first worksheets are transferred. If the worksheets are not automatically paired, you can switch to "Manual".

Manually pair worksheets

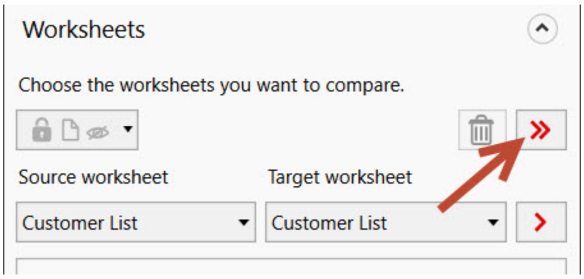
The worksheets can also be paired manually. This can be done in the header of the worksheets list.



Above the worksheets table, there are two drop-down list boxes (see chart). Using these boxes, the desired worksheets can be selected. Select the desired worksheets and confirm the selection with the arrow button to the right. The worksheet pair will then be added to the worksheets list.

Match all worksheets

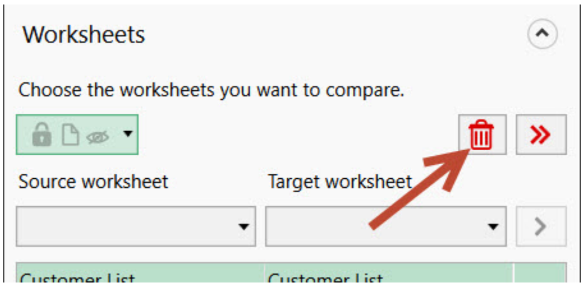
You can also match all worksheets. This is done using the following button:



All equally named worksheets will be paired.

Remove worksheet pairs

If you do not want to compare all worksheet pairs, you can empty the worksheets list using the "Delete" button.



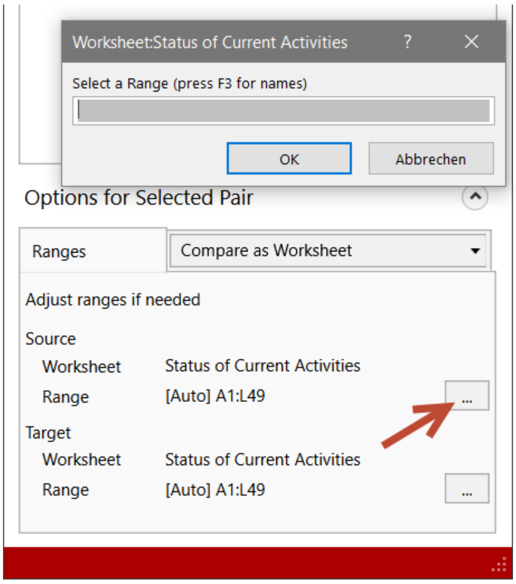
Choose the worksheets you want to compare.

As a further option, the unwanted worksheet pairs can be removed by selecting the pairs and clicking on the "Delete" button.

No worksheets found

If no worksheets are available in the drop-down lists, the worksheets are either protected, empty and/or hidden. In this case, you need to change the display mode for worksheets. For more details please refer to the chapter "Automatically pair worksheets".

3.2.2 Ranges


Worksheet:Status of Current Activities
?
OK
Abbrechen
Options for Selected Pair
Ranges
...
...

Select a Range (press F3 for names)

Compare as Worksheet

Adjust ranges if needed

Source
Worksheet Status of Current Activities
Range [Auto] A1:L49

Target
Worksheet Status of Current Activities
Range [Auto] A1:L49

If you do not want to compare the entire worksheet, you can define the desired range. Proceed as follows:

1. Select the worksheet which contains the range you want to compare
2. Under the worksheet pair the options area appears (3 options). Now you can specify the desired range by using of the range buttons.
3. The range address is also listed next to the worksheet name.

Note:
You can also enter named ranges. Multiple ranges are not allowed.

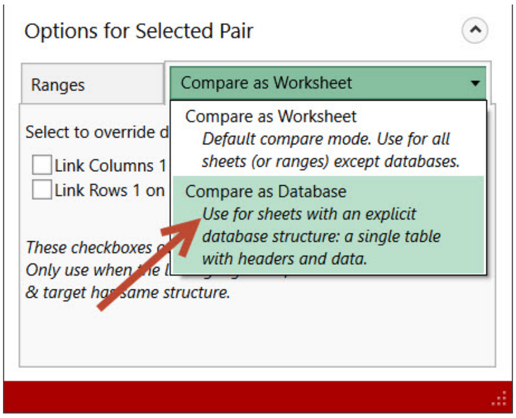
3.2.3 Databases

If your worksheets contain a database structure, then we recommend a database comparison. If you do not know exactly what a database is and how it works, please read the chapter "General Information About Databases" first.

Proceed as follows to enter the database options:

1. Select the worksheet pair which contains a database structure.

2. Click with the mouse on the check box "Compare as a database".



Options for Selected Pair

Ranges | Compare as Worksheet ▾

Select to override d

□ Link Columns 1

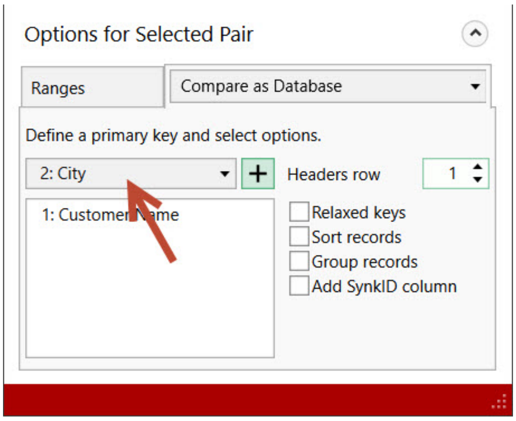
□ Link Rows 1 on

Compare as Worksheet
Default compare mode. Use for all sheets (or ranges) except databases.

Compare as Database
Use for sheets with an explicit database structure: a single table with headers and data.

These checkboxes o
Only use when the l
& target has same structure.

4. The tab "Database settings" appears. Now you have to define the primary key by which the databases will be compared. To do this, select the necessary columns and confirm your selection with the Plus button.



Options for Selected Pair

Ranges | Compare as Database ▾

Define a primary key and select options.

2: City ▾ + Headers row 1

1: Customer Name

□ Relaxed keys

□ Sort records

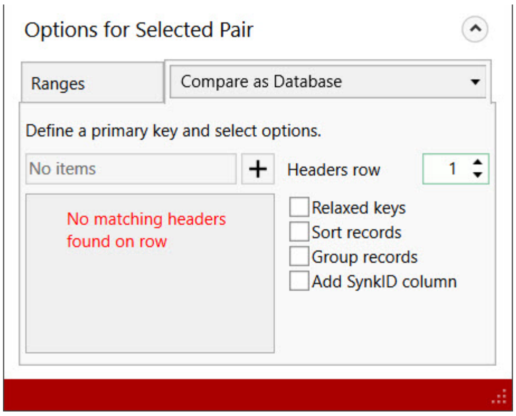
□ Group records

□ Add SynkID column

Important:

The field names must be identical, otherwise the database can not be compared!

If no identical field name is, the following message appears:



Options for Selected Pair

Ranges | Compare as Database

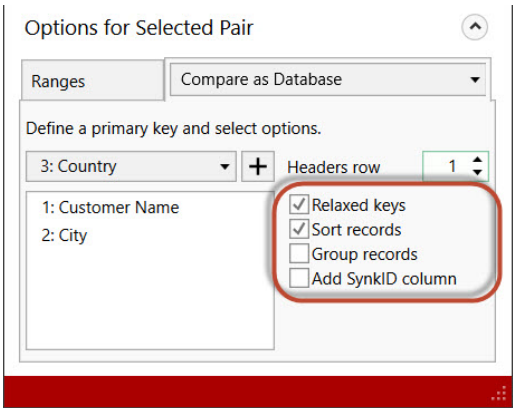
Define a primary key and select options.

No items | Headers row 1

No matching headers found on row

□ Relaxed keys
□ Sort records
□ Group records
□ Add SynkID column

5. Finally, you can set the database options:



Options for Selected Pair

Ranges | Compare as Database

Define a primary key and select options.

3: Country | Headers row 1

1: Customer Name
2: City

☑ Relaxed keys
☑ Sort records
□ Group records
□ Add SynkID column

The meaning of the database options are as follows:

Relaxed keys

When selecting this option, differences in spaces and upper/lower case are ignored.

By default the primary keys are matched case sensitive (e.g. "John Miller" and "JOHN MILLER ").

Sort records

The records are sorted by the primary key.

Group records

The data are grouped. The date records are grouped as follows:

1. Linked records

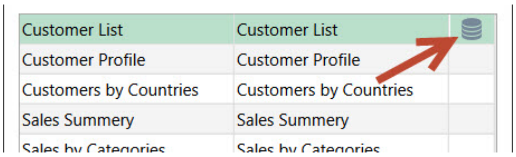
2. New/deleted records
3. Duplicate keys
4. Duplicate records (redundant records)

Add SynkID column

At the end of each database record, a new column called "SynkID" is appended, which contains the difference types. You can then sort the databases by the SynkID and/or filter them (auto filter). The following IDs are written in the SynkID column:

0:OK	Row with no differences
1:DIFF	Row with differences
2:MISS	Missed row
3:DUPKEY	Duplicate keys
4:DUPREC	Duplicate records (redundant records)

6. When you have completed the database settings, a database icon appears in the worksheet list next to the worksheet pair.



Customer List	Customer List	🛢
Customer Profile	Customer Profile	
Customers by Countries	Customers by Countries	
Sales Summery	Sales Summery	
Sales by Categories	Sales by Categories	

Notes:
The maximum length of a primary key field is limited to 64 characters. If you should operate with primary key fields with more than 64 characters, then unwanted duplicate keys can occur. If possible do not operate with long primary key fields.

3.2.3.1 General Information About Databases

Prior to describing the types of differences, we want to give a brief comparison of "normal" worksheets versus databases.

"Normal" worksheets

do not contain column headings and no database structure. For such worksheets, Synkronizer performs a cell-by-cell comparison. As a frequent user of Synkronizer you will soon learn that the data comparison of a "normal" worksheets is not as efficient as a database comparison.

Databases

are worksheets that contain data within a database structure. The data must be arranged as follows:

- The first row contains **unique column names**.
- The database contains one (or more) column(s) with **primary key(s)** (unique identification number, article number etc.). A primary key uniquely identifies each data record and facilitates the sorting and synchronizing of your data. The primary key does not need to be a single field (e.g. employee ID). It may consist of multiple fields (e.g. family name, first name and city) to create a unique identifier and avoid duplicate keys/records.
- The data must be **sortable** and can not contain any merged cells.
- The table can't contain formulas that refer to other rows. Formula references to other columns within the same row are allowed.

Following paragraphs describe the difference between a 'duplicate key' and 'duplicate records'.

Duplicate keys

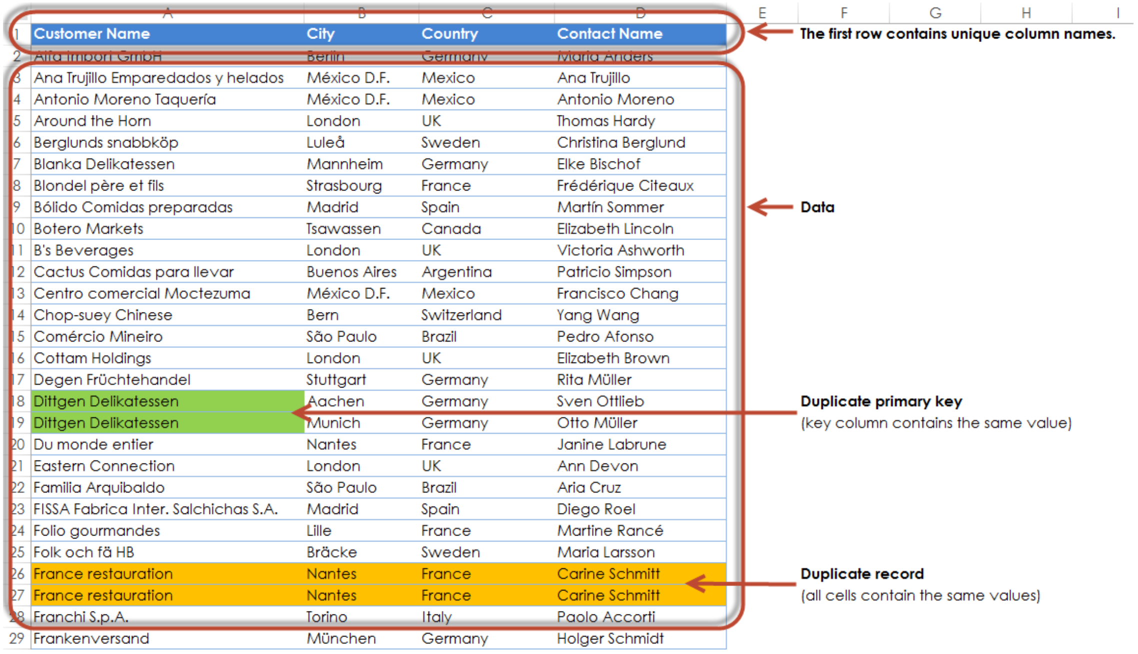
Duplicate keys occur, when data records contain the same primary key. Assuming the primary key is formed from the family and first name, and given you have several customers called "John Miller" in your database, Synkronizer will not know which data records to compare. So make sure you operate with unique primary keys when working with Synkronizer.

Duplicate Records (Redundant records)

Synkronizer also checks whether your database contains duplicate records (or redundant records). These are records where all fields (not just the key fields) are equal. If Synkronizer finds any of these records, they will be highlighted in gray. Duplicate records do not serve any purpose and should be deleted.

The comparison of "databases" is highly efficient. The databases may be sorted or altered. Inserting and deleting of columns and rows is allowed. Synkronizer will recognize all differences between the two files.

Scheme Database



	A	B	C	D
1	Customer Name	City	Country	Contact Name
2	Alfa Import GmbH	Berlin	Germany	Maria Anders
3	Ana Trujillo Emparedados y helados	México D.F.	Mexico	Ana Trujillo
4	Antonio Moreno Taquería	México D.F.	Mexico	Antonio Moreno
5	Around the Horn	London	UK	Thomas Hardy
6	Berglunds snabbköp	Luleå	Sweden	Christina Berglund
7	Blanka Delikatessen	Mannheim	Germany	Elke Bischof
8	Blondel père et fils	Strasbourg	France	Frédérique Citeaux
9	Bólido Comidas preparadas	Madrid	Spain	Martín Sommer
10	Botero Markets	Tsawassen	Canada	Elizabeth Lincoln
11	B's Beverages	London	UK	Victoria Ashworth
12	Cactus Comidas para llevar	Buenos Aires	Argentina	Patricio Simpson
13	Centro comercial Moctezuma	México D.F.	Mexico	Francisco Chang
14	Chop-suey Chinese	Bern	Switzerland	Yang Wang
15	Comércio Mineiro	São Paulo	Brazil	Pedro Afonso
16	Cottam Holdings	London	UK	Elizabeth Brown
17	Degen Früchtehandel	Stuttgart	Germany	Rita Müller
18	Dittgen Delikatessen	Aachen	Germany	Sven Ottlieb
19	Dittgen Delikatessen	Munich	Germany	Otto Müller
20	Du monde entier	Nantes	France	Janine Labrune
21	Eastern Connection	London	UK	Ann Devon
22	Familia Arquibaldo	São Paulo	Brazil	Aria Cruz
23	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain	Diego Roel
24	Folio gourmandes	Lille	France	Martine Rancé
25	Folk och fä HB	Bräcke	Sweden	Maria Larsson
26	France restauration	Nantes	France	Carine Schmitt
27	France restauration	Nantes	France	Carine Schmitt
28	Franchi S.p.A.	Torino	Italy	Paolo Accorti
29	Frankenversand	München	Germany	Holger Schmidt

The first row contains unique column names.

Data

Duplicate primary key
(key column contains the same value)

Duplicate record
(all cells contain the same values)

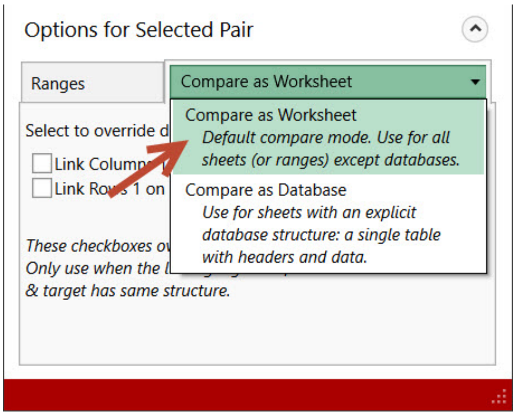
3.2.4 Link Data 1 on 1

If you are sure that your worksheets do not contain new or deleted rows/columns, then you can compare your files with the "1 on 1 method". In this case, the rows and columns are compared "1 on 1" and only cell differences will be detected.

Proceed as follows to compare the rows and/or columns by the "1 on 1 linking method":

1. Select the worksheet pair, which should be compared 1 on 1.

2. Click "Compare as worksheet". The tab "Linking Options" appears.



Options for Selected Pair

Compare as Worksheet
Default compare mode. Use for all sheets (or ranges) except databases.

Compare as Database
Use for sheets with an explicit database structure: a single table with headers and data.

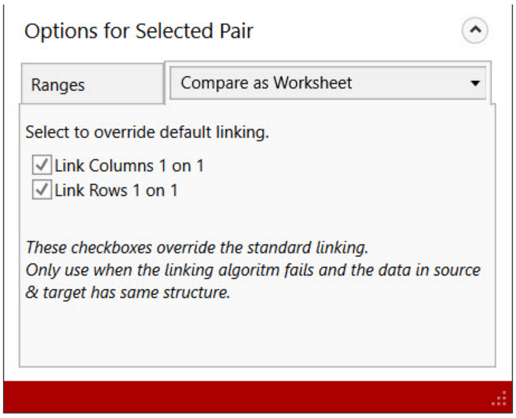
Select to override d

□ Link Column

□ Link Rows 1 on

These checkboxes o
Only use when the l
& target has same structure.

4. The tab "Linking Options" appears. Click "Link columns 1 on 1" or "Link rows 1 on 1"



Options for Selected Pair

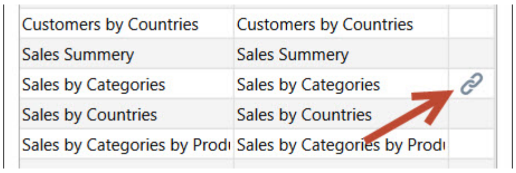
Ranges: Compare as Worksheet

Select to override default linking.

☑ Link Columns 1 on 1
☑ Link Rows 1 on 1

These checkboxes override the standard linking.
Only use when the linking algoritm fails and the data in source & target has same structure.

5. After confirming the 1 on 1 linking method, the linking icon appears next to the worksheet pair.



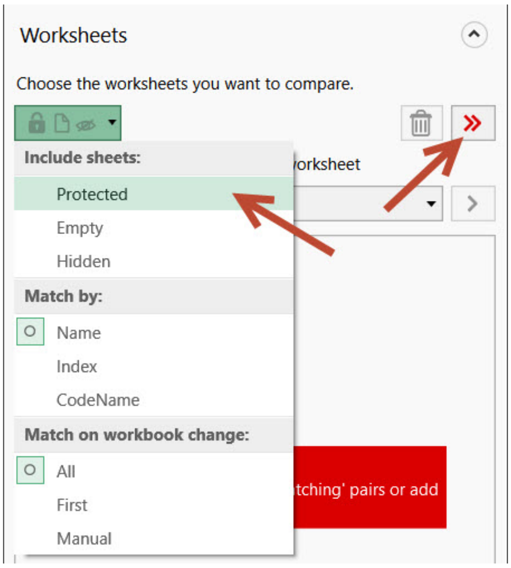
Customers by Countries	Customers by Countries	
Sales Summery	Sales Summery	
Sales by Categories	Sales by Categories	🔗
Sales by Countries	Sales by Countries	
Sales by Categories by Prod	Sales by Categories by Prod	

3.2.5 Passwords

With Synkronizer password-protected worksheets can also be compared. The disadvantage is that password-protected worksheets are not updated and that the differences can not be marked or outlined. But if you know the password/passwords of the worksheet(s), you can unlock the worksheets as follows:

1. Open the files which contain password-protected worksheets.

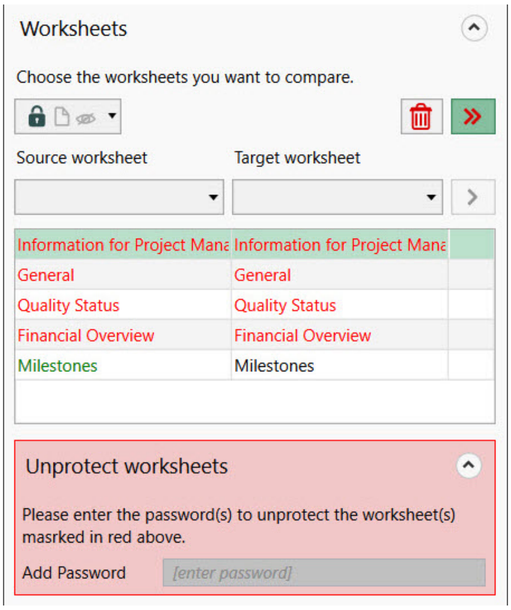
2. If the worksheets should not appear in the worksheets list, you might need to enable the "Protected" option in the pairing worksheet element:



Choose the worksheets you want to compare.

Select "Protected" and click the "Add worksheets" button - the protected worksheets are now displayed.

3. The password-protected worksheets are displayed in green (successfully unlocked worksheet) or red (protected worksheet) and a red area, in which you can enter the password to unlock the worksheets:



Choose the worksheets you want to compare.

Source worksheet	Target worksheet
Information for Project Mana	Information for Project Mana
General	General
Quality Status	Quality Status
Financial Overview	Financial Overview
Milestones	Milestones

Please enter the password(s) to unprotect the worksheet(s) masrked in red above.

4. Enter the password in the field a and unlock the worksheets. If you have successfully entered all the passwords, the worksheets are shown in green.

Passwords in project files

The worksheet passwords are stored coded in the projects.

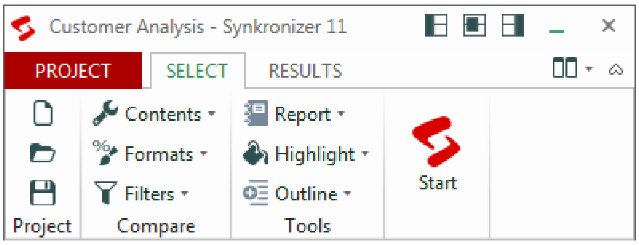
Note:

When comparing protected sheets without supplying a valid password please be aware of the following:

If the sheet contains hidden formulas (Cell Properties/Protection/Hidden) the formulas for the sheet cannot be read and the sheet's values are used instead.

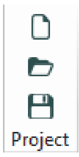
3.3 Ribbon

In the upper part of the selection form you'll find a toolbar which allows you to define numerous options:



The following options are available:

Project options

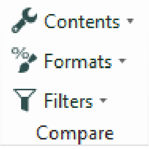


Reset project. All settings are set back to defaults

Load a project

The current comparison settings (files, worksheets, format and filter settings) are saved as a project. Any passwords are not saved in the project file.

Compare options

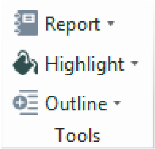


Compare additional content like comments and Excel names.

Define format options (colors, fonts, number formats, etc.)

Define filters

Tools options

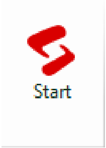


Create a difference report

Define highlight options

Define outline settings / show/hide options

Start



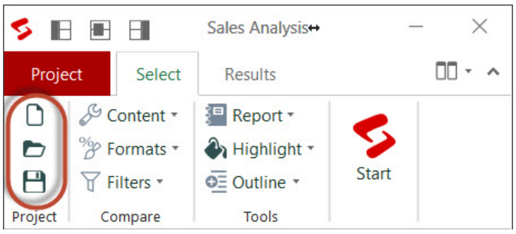
Start the comparison

3.3.1 Project

With the "Project" commands you have the option of saving recurring comparison processes as a project. Assuming you always have to compare the same files with each other, you can save this comparison process (s) as a project. Next time just open this project and you can start comparing right away.

The following data is saved in the project file:

- file names
- worksheet names
- work settings
- database settings
- formats
- filters
- Passwords are **not** saved



New

The form is cleared, all settings are reset.

Open

This command opens an existing project.

Save

The current comparison settings (files, worksheets, format and filter settings) are saved as a project. Any passwords are not saved in the project file.

The projects can also be managed in the Project register.

3.3.2 Compare - Content

In this menu, you can choose whether additional content needs to be compared. The following content options can be selected:

- **Comments**
  (Cell) Comments are information that can be added to each cell.

- **Names**
  Names are placeholders which can be used to identify a range of cells, a function, a constant, or a table.

Proceed as follows to perform the comparison of additional content.
1. Click Contents.

2. Toggle comments/names according to your needs.

3. When activated, the contents/names icons change from gray to black.

3.3.3 Compare - Formats

The following cell formats can be compared:

- **Number**
  It is compared whether the number format of the cells is different.

- **Alignment**
  Compares the alignment of the cells (left, center, etc.)

- **Font**
  All the font formattings are compared (font name, size, color, underline, etc.)

- **Border**
  The cell borders are compared.

- **Fill**
  The background color of the cells is compared.

- **Protection**
  The cell protection and visibility of the cell to be compared.

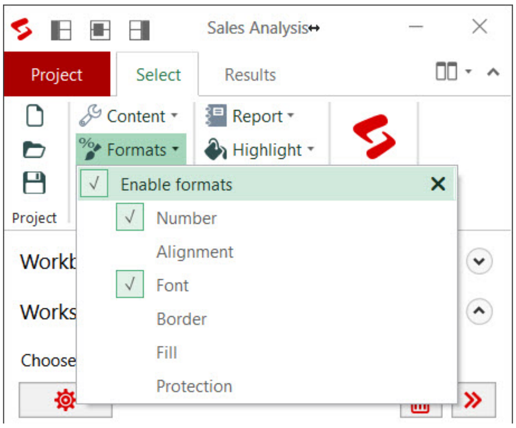
Proceed as follows to compare the cell formats:

1. Click on formats.

2. Select "Enable formats". (The format icon changes from gray to black).

3. Select all of the formats that you want to compare.

In this example, the number formats and font sizes are compared.



☑ Enable formats

☑ Number

Alignment

☑ Font

Border

Fill

Protection

3.3.4 Compare - Filters

The filtering function enables you to reduce the number of differences that will be found and displayed. Just click "Enable filters" and choose the filter(s) which you want to apply.

- **Ignore case**
  If this box is checked, differences in upper and lower case writing are ignored (e.g. "John Wayne" equals to "JOHN WAYNE").

- **Ignore whitespace**
  If this box is checked, spaces are ignored at the beginning and end of words, e.g. the values ? "John Wayne" and " John Wayne " (additional spaces at the beginning and at the end) are considered as equal.

- **Ignore data type**
  IIt is possible that some numbers are formatted as number and others as text. This will logically be evaluated as a difference, although the numbers are identical. With this option, these differences are ignored.

- **Ignore formulas with same results**
  Supposed you have two different formulas which contain the same results. See the following example:

  Cell source file
  Formula: =D50
  Result: 100

  Cell target file
  Formula: =$D$50
  Result: 100

  These types of differences are filtered out by activating this option.

- **Ignore calculated values**
  All cells that contain identical formulas, are ignored.

- **Ignore constants**
  If this option is active, all cells with constants (entered values)? are ignored.

- **Ignore formulas**
  All formula fields are filtered out .

- **Ignore hidden columns**
  All columns that are hidden in the source file, are ignored. Hidden columns of the target file are compared.

- **Ignore hidden rows**
  Works like "hidden columns", but with rows.

- **Numeric tolerance (#) / Relative tolerance (%)**
  For numeric fields, you can instruct Synkronizer how to deal with deviations. Supposed only deviations greater than 0.10 need to be shown, then enter 0.10. Deviations up to and

including 0.10 will then be ignored. You can distinguish between numeric and relative tolerances. Just click on the label 'Numeric tolerance' to switch the tolerance type.

- **Equivalent values**
  Chances are, that in some worksheets the same difference shows up en masse. Suppose you need to compare checklists that have a column called "Completed". In the first checklist, the values in this column are represented by "yes / no", but in the other checklist by "ja / nein" or "1/0". This would cause a large number of false positives to be reported in a normal comparison. You can now filter out these differences by defining the corresponding equivalents, e.g. 1=yes, 1=ja, yes=ja, 0=no, 0=nein, no=nein etc. If you want to filter empty cells, enter [empty] or leave the field blank.

  You can also use the following **wildcards**:

  ? = any character
  * = any number of characters

  For example, if you enter "H?llo" and "H?llo", the values Hello, Hello and Hollo will all be recognized as identical.
  If you enter e.g. "good *" and "good *", then the good morning, good evening and good afternoon will all be recognized as identical.

Proceed as follows to activate one or more filters :

1. Click Filters.

2. Select "Enable filters". (The filter icon changes from gray to black).

3. Select the filter(s) that you want to activate.

4. Close the filter window .

3.3.5 Tools - Highlight

In this menu, you can choose whether the differences are to be marked in the underlying Excel spreadsheets or not. The following options are available:

- **No highlighting / Undo highlighting**
  The Excel worksheets remain unchanged.

- **Highlight differences**
  All differences are highlighted in the underlying Excel worksheets.

- **Clear & highlight differences**
  All differences are highlighted in the underlying Excel worksheets. Additionally, the background colors of the Excel worksheets are deleted prior to the comparison.

Depending upon your choice, the check icon is displayed gray or black. If the differences are to be marked, the symbol is black, otherwise gray.

3.3.6 Tools - Report

In the Report menu you can choose whether an additional difference report should be generated or not. The following options are available:

- **No report**
  No difference report will be generated.

- **Standard report**
  A "normal" difference report will be created which contains all differences.

- **Hyperlinked report**
  A difference report will be generated which contains all differences and each difference holds a hyperlink pointing to the corresponding difference.

- **XML report (workbook)**
  An XML report will be created. All differences are listed in a clear table.

- **XML report (XML only)**
  The differences are stored in an XML file. The XML file is stored in the following directory:
  » ThisPC » Documents »Synkronizer Reports

The report icon is displayed either gray or black depending on its status. If a difference report is selected, the symbol is black, otherwise gray.

3.3.7 Tools - Outline

In the Outline menu, you can choose whether you want to see only certain types of differences or not. In this way you can check and process the differences very efficiently. The following difference types can be selected:

- **Missing columns**
  These columns are only found in one worksheet.

- **Missing rows**

These rows were either deleted or added.

- **Duplicate rows**
  Applies only to database mode: these rows include either duplicate (redundant) records or rows with duplicate keys.

- **Identical rows**
  These rows are completely identical and have no cell differences.

- **Different rows**
  These rows contain cell differences.

Proceed as follows if only certain differences need to be displayed:

1. Click Outline

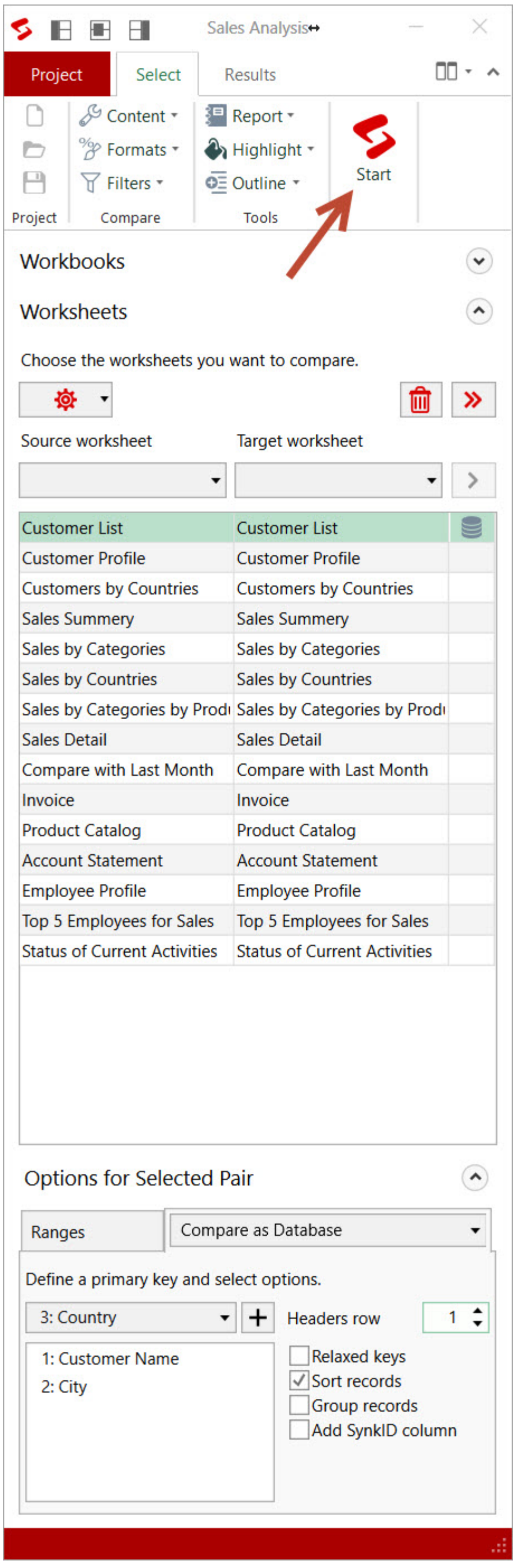
2. Select "Enable outlining". (The outline symbol will change from gray to black)

3. Select the types of difference, which you want to view.

4. Close the dialogue "Outline".

3.3.8 Start - Compare files

After you have selected all the worksheets, settings and comparison options, you can start comparing the worksheets. Just click on the Start icon and the files will be compared.



Choose the worksheets you want to compare.

Source worksheet	Target worksheet
Customer List	Customer List
Customer Profile	Customer Profile
Customers by Countries	Customers by Countries
Sales Summery	Sales Summery
Sales by Categories	Sales by Categories
Sales by Countries	Sales by Countries
Sales by Categories by Prodi	Sales by Categories by Prodi
Sales Detail	Sales Detail
Compare with Last Month	Compare with Last Month
Invoice	Invoice
Product Catalog	Product Catalog
Account Statement	Account Statement
Employee Profile	Employee Profile
Top 5 Employees for Sales	Top 5 Employees for Sales
Status of Current Activities	Status of Current Activities

Options for Selected Pair

Ranges: Compare as Database

Define a primary key and select options.

3: Country

Headers row: 1

1: Customer Name
2: City

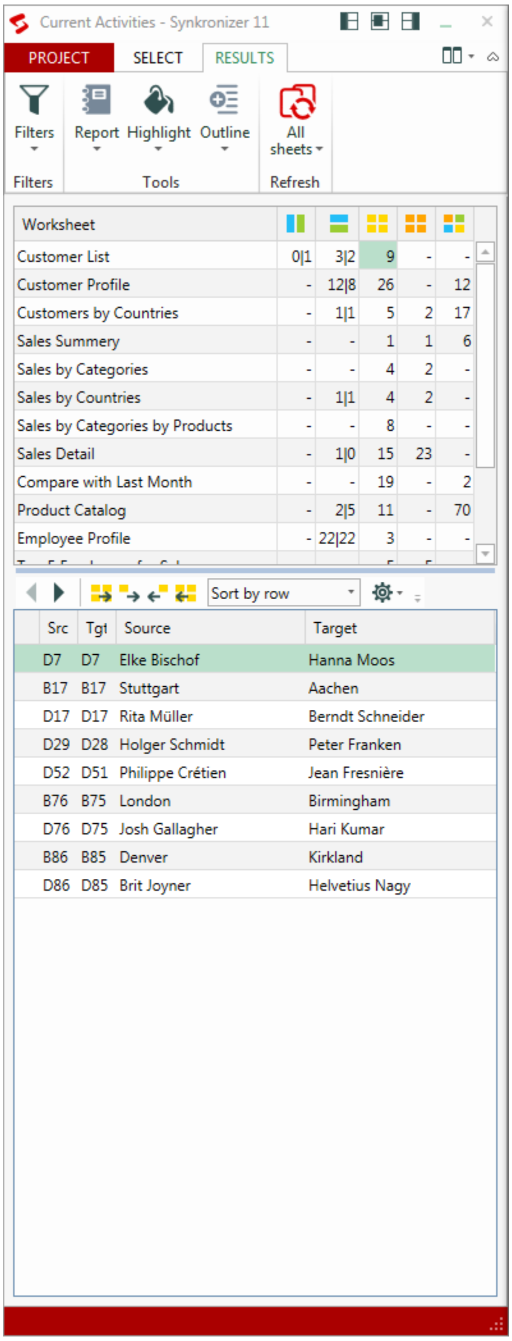
□ Relaxed keys
☑ Sort records
□ Group records
□ Add SynkID column

After comparison, the files and the results form will be displayed side by side. With the results form, you can then smartly check the differences and eliminate them.

Part IV

4 Results

After comparing, the results form as well as the worksheets are arranged side by side. In the results form, you can then smartly check and eliminate any difference.



Worksheet	▥	▤	▦	▦	▦
Customer List	0|1	3|2	9	-	-
Customer Profile	-	12|8	26	-	12
Customers by Countries	-	1|1	5	2	17
Sales Summery	-	-	1	1	6
Sales by Categories	-	-	4	2	-
Sales by Countries	-	1|1	4	2	-
Sales by Categories by Products	-	-	8	-	-
Sales Detail	-	1|0	15	23	-
Compare with Last Month	-	-	19	-	2
Product Catalog	-	2|5	11	-	70
Employee Profile	-	22|22	3	-	-

Src	Tgt	Source	Target
D7	D7	Elke Bischof	Hanna Moos
B17	B17	Stuttgart	Aachen
D17	D17	Rita Müller	Berndt Schneider
D29	D28	Holger Schmidt	Peter Franken
D52	D51	Philippe Crétien	Jean Fresnière
B76	B75	London	Birmingham
D76	D75	Josh Gallagher	Hari Kumar
B86	B85	Denver	Kirkland
D86	D85	Brit Joyner	Helvetius Nagy

Ribbon

In this section, various actions can be performed. With these actions, you can selectively and efficiently review and manage the differences.

Summary

This pane shows the worksheets and the number of differences found.
Differences for the selected worksheet/category are shown in the lower pane.

Differences

This pane shows each difference found for the selected worksheet/category cell in the upper pane. You can navigate to each difference and decide if the difference needs to be transferred or not.

Synkronizer will automatically display the corresponding areas of both worksheets side by side, so you can easily and directly overview the differences. At this point, you may decide with which version you wish to continue. You can transfer values from the source sheet to the target sheet or vice versa. This is the most efficient way to eliminate all differences, and your Excel worksheet will be perfectly updated in no time.

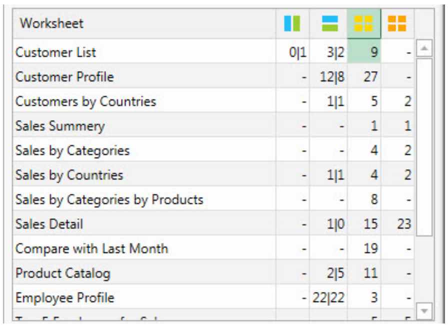
Actions

With the results form, you can perform the following actions:

- Overview of differences
- Synchronize worksheets / transfer differences
- Delete unwanted differences
- Show/hide differences

4.1 Summary

The upper part of the Results form displays the worksheets and their differences. You can see the number of differences that were found between the worksheets at a glance.



Worksheet	▐▌	〓	⊞	⊞
Customer List	0|1	3|2	9	-
Customer Profile	-	12|8	27	-
Customers by Countries	-	1|1	5	2
Sales Summery	-	-	1	1
Sales by Categories	-	-	4	2
Sales by Countries	-	1|1	4	2
Sales by Categories by Products	-	-	8	-
Sales Detail	-	1|0	15	23
Compare with Last Month	-	-	19	-
Product Catalog	-	2|5	11	-
Employee Profile	-	22|22	3	-

In the columns of the list box, the various differences are displayed. In this screenshot, for example, the following differences were found in the worksheet "Customer List":

Missing columns	0|1; (no different column in source worksheet, one different column in target worksheet)
Missing rows	3|2; (three different rows in source worksheet, two different rows in target worksheet)
Differences in entered values/formulas	9 differences in entered values/formulas
Differences in calculated formulas	0

The following difference types are reported in this list box:

▐▌ Missing columns. These columns exist only in one worksheet.

〓 Duplicate records (redundant rows). These records are completely identical. These records should be deleted. (Appears only in the database comparison mode.)

〓 Duplicate keys. These records have the same primary key. (Appears only in the database comparison mode.)

〓 Missing rows. These rows exist only in one worksheet.

- Differences in entered cell values and/or formulas. The entered cell values and/or formulas are different in the underlying worksheets.

  Note:
  If you compare protected tables and a table contains protected and hidden cells, then only the cell values are compared. Different cell formulas are not recognized in this case.

- Differences in calculated cell formulas. The results of the cell formulas are different (the entered cell formulas are identical).

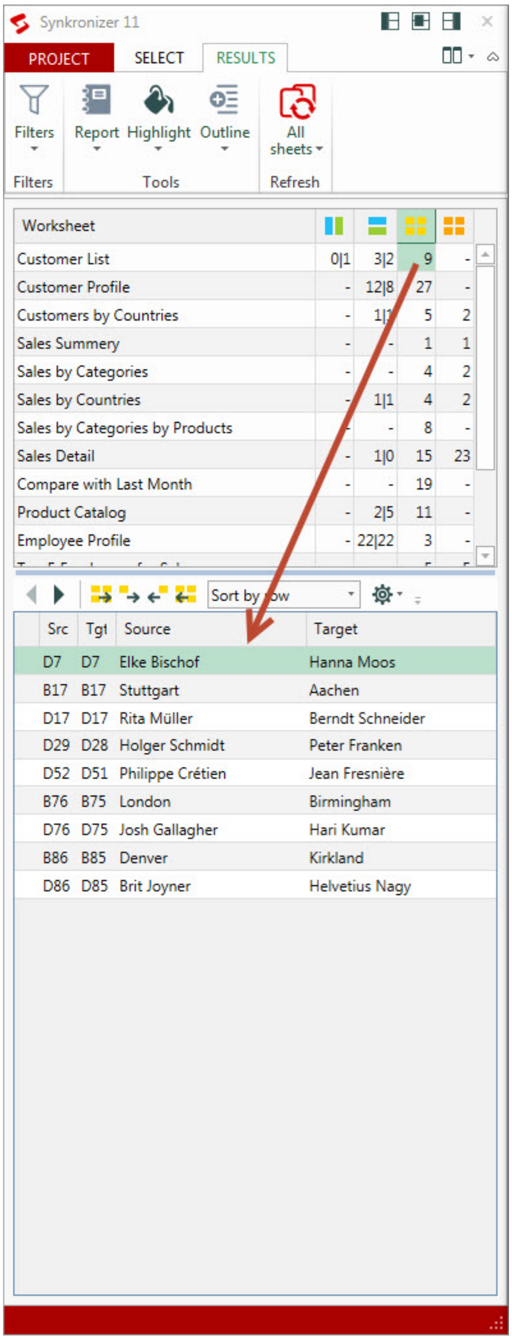
- Different cell formats. (Appears only when formats are compared.)

- Different comments. (Appears only when comments are compared.)

- Different names. (Appears only when names are compared.)

For each difference, the detailed information can be displayed in the lower part of the form. Just click with the mouse on the desired type of difference and you will find further details below.

In the following example, the difference type "Entered values & formulas" is selected. Nine differences were found in total (in the upper part of the form). The detailed information of each difference is displayed in the lower part of the form:

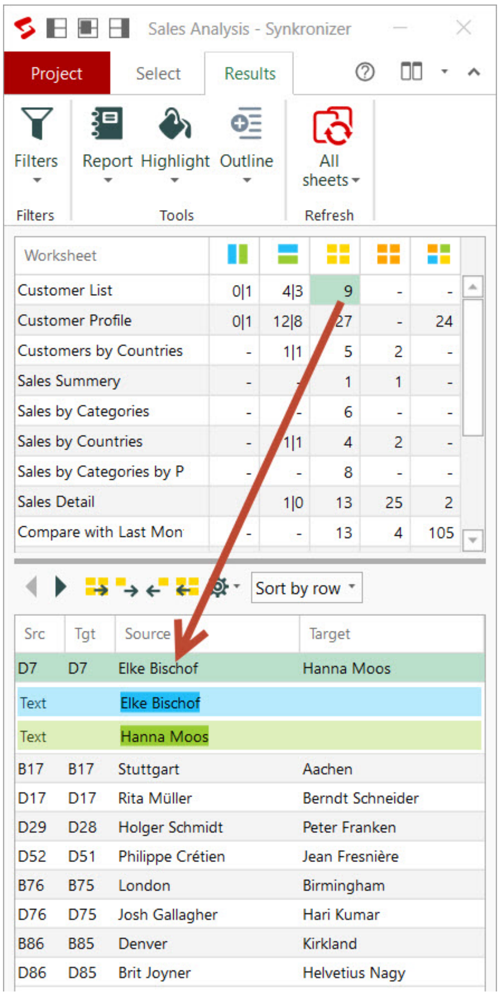


Worksheet				
Customer List	0|1	3|2	9	-
Customer Profile	-	12|8	27	-
Customers by Countries	-	1|1	5	2
Sales Summery	-	-	1	1
Sales by Categories	-	-	4	2
Sales by Countries	-	1|1	4	2
Sales by Categories by Products	-	-	8	-
Sales Detail	-	1|0	15	23
Compare with Last Month	-	-	19	-
Product Catalog	-	2|5	11	-
Employee Profile	-	22|22	3	-

Src	Tgt	Source	Target
D7	D7	Elke Bischof	Hanna Moos
B17	B17	Stuttgart	Aachen
D17	D17	Rita Müller	Berndt Schneider
D29	D28	Holger Schmidt	Peter Franken
D52	D51	Philippe Crétien	Jean Fresnière
B76	B75	London	Birmingham
D76	D75	Josh Gallagher	Hari Kumar
B86	B85	Denver	Kirkland
D86	D85	Brit Joyner	Helvetius Nagy

4.2 Differences

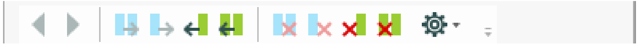
In the lower part of the form, the details of the difference selected in the upper part are displayed.



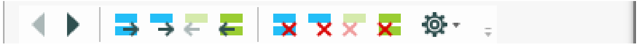
Worksheet					
Customer List	0|1	4|3	9	-	-
Customer Profile	0|1	12|8	27	-	24
Customers by Countries	-	1|1	5	2	-
Sales Summery	-	-	1	1	-
Sales by Categories	-	-	6	-	-
Sales by Countries	-	1|1	4	2	-
Sales by Categories by P	-	-	8	-	-
Sales Detail		1|0	13	25	2
Compare with Last Mon	-	-	13	4	105

Src	Tgt	Source	Target
D7	D7	Elke Bischof	Hanna Moos
Text		Elke Bischof	
Text		Hanna Moos	
B17	B17	Stuttgart	Aachen
D17	D17	Rita Müller	Berndt Schneider
D29	D28	Holger Schmidt	Peter Franken
D52	D51	Philippe Crétien	Jean Fresnière
B76	B75	London	Birmingham
D76	D75	Josh Gallagher	Hari Kumar
B86	B85	Denver	Kirkland
D86	D85	Brit Joyner	Helvetius Nagy

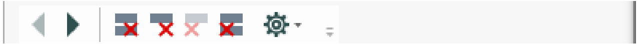
Toolbar for Missing Columns



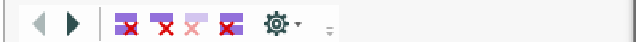
Toolbar for Missing Rows



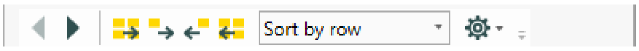
Toolbar for Duplicate Database Records



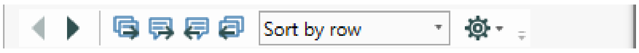
Toolbar for Duplicate Database Keys



Toolbar for Different Cells



Toolbar for Different Comments



All differences can be processed, transferred or deleted with the following keys:

Select difference

With these buttons you can navigate to the next or previous difference.

Transfer different columns/rows from source to target worksheet

With these buttons, the columns/rows are copied from the source to the target worksheet (from left to right). It is possible to copy just the selected column/row or all columns/rows. Once the columns/rows have been copied, they are displayed in red.

These commands are also available in columns that contain at least one difference.

Transfer different columns/rows from target to source worksheet

As above, but in the other direction.

Delete different columns/rows in the source worksheet

With these buttons, the columns/rows in the source worksheet (left) will be deleted. It is either possible to delete all or only the selected column/row.

Delete different columns/rows in the target worksheet

As above

Transfer cell differences/comments from source to target worksheet

With these buttons, the different cell values or comments are copied from the source to the target worksheet (from left to right). It is either possible to copy all or only the marked difference. Once the difference has been copied, it is shown in red.

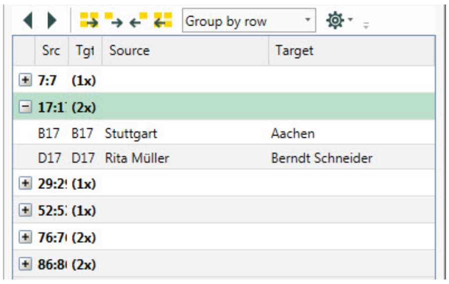
Transfer cell differences/comments from target to source worksheet

As above.

Sort by row With this button, the sorting of cell differences can be changed. As required, the cell differences can be sorted or grouped differently. The cell differences are by default sorted by rows and within the rows by columns.

Sort by column The cell differences are sorted first by columns and within columns to rows.

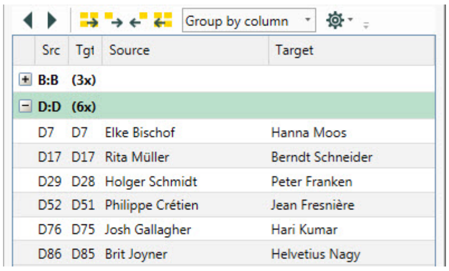
Group by row If you choose this option, the differences are grouped by rows. The differences can be transferred individually per line or in groups.



	Src	Tgt	Source	Target
⊞	7:7	(1x)		
⊟	17:1	(2x)		
	B17	B17	Stuttgart	Aachen
	D17	D17	Rita Müller	Berndt Schneider
⊞	29:2	(1x)		
⊞	52:5	(1x)		
⊞	76:7	(2x)		
⊞	86:8	(2x)		

Group by column

If you choose this option, the differences are grouped by columns. The differences can be transferred individually per line or in groups.



	Src	Tgt	Source	Target
⊞	B:B	(3x)		
⊟	D:D	(6x)		
	D7	D7	Elke Bischof	Hanna Moos
	D17	D17	Rita Müller	Berndt Schneider
	D29	D28	Holger Schmidt	Peter Franken
	D52	D51	Philippe Crétien	Jean Fresnière
	D76	D75	Josh Gallagher	Hari Kumar
	D86	D85	Brit Joyner	Helvetius Nagy

This option is extremely useful when you are working with databases and need to update cell differences of a column. With just one click, all the differences of a column can be transferred.

Settings

Depending on the difference type, different options can be set, which are described below.

- ### Updating options
  This option is only applicable for missing columns and rows.
  Here you can specify which data and formats should be transferred. The following options are available:

  basic: only cell values and the cell alignment (left, right, centered) are transferred
  Medium: additionally the background color is transferred
  full: all data and cell formats are transferred

  This setting is especially important for large Excel files, because the data transfer can be very slow for large data.

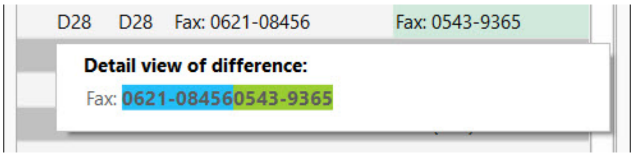
- ### Insert entire row or column
  This option is only important if you are working with cell ranges. Normally,

when you insert a row, the row will be also inserted outside the cell range. By disabling this option, the rows are inserted only within the cell range. This option is only applicable for missing columns and rows.

- **Show details of cell differences**
  Here you can determine whether the detailed information of the cell differences is to be displayed inline in the grid or as popup. By default, the differences are displayed inline in the grid.

- **Empty cells may overwrite data**
  If this box is checked, cells holding a value will be overwritten by empty cells, so that the previous value is lost. If this is not desired, you can disable the check box.

Detail view of text difference



D28 D28 Fax: 0621-08456 Fax: 0543-9365

Detail view of difference:

Fax: 0621-084560543-9365

In the case of text differences (yellow symbol, "Differences in entered cell values ?and/or formulas"), you can move the mouse over the desired difference (do not click), whereupon the detailed view of the text difference is displayed as a tooltip.

Differences displayed with an "at" sign (@)

If a number, date or Boolean value (true/false) is formatted as text, the corresponding differences will be displayed with an "at" sign (@). E.g., the number 123 formatted as text will be displayed as "@:123".

4.3 Ribbon

In addition, various actions can be performed in the results form. With these actions, you can selectively and efficiently manage the differences.

Filter

If necessary, you can define a (new) filter and compare anew. This way, you can immediately see whether you get better comparison results or not. Proceed as follows to create filter and start a new comparison:

1. Click on the arrow below the filter icon.
2. Select "Enable filters" (the filter icon changes).
3. Click on any filter that you want to enable or disable.
4. Close the filters dialogue.
5. Click the "Refresh" icon.
6. The worksheet(s) is/are newly compared.

Report

You can create a difference report at any time. Proceed as follows:

1. Click the arrow below the report icon.
2. Select the report you want to create.
3. The report is generated.

Highlight

Here you can choose whether only certain types of difference are to be shown. Proceed as follows to display only certain types of difference:

Outline

Here you can decide whether only certain types of difference are to be shown. Proceed as follows to display only certain types of difference:

1. Click the arrow below the Outline icon.
2. Select "Enable outlining" (the Outline icon changes)
3. Click on the difference types to be displayed.
4. Close the Outline dialogue.
5. Click on the Outline icon.
6. Only the selected different types will be displayed.

Refresh

By clicking on this icon, the files are compared anew. You can choose whether all worksheets or only the active worksheet shall be re-compared.

Part V

5 Install, Uninstall

Synkronizer 11 is a COM addin for Excel 2016 (or 365), 2013 , 2010 and 2007. Synkronizer works both in 32-, and 64-bit mode.

Installation

1. Download the latest version.

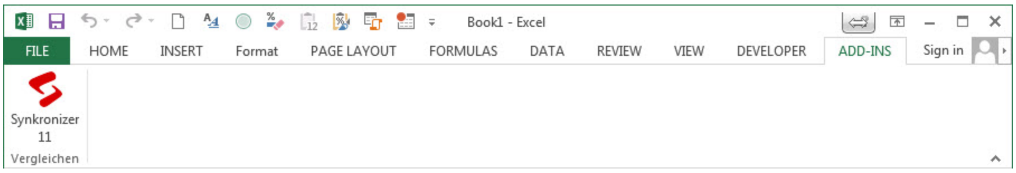
2. Log on to Windows with administrator privileges.

3. Close Excel.

4. Open the the downloaded zip file, double click on the setup file and follow instructions...

5. Start Excel.

6. Synkronizer can now be started from the Add-Ins Tab



7. Register the software.

Versions & Updates

The installer will automatically uninstall older builds of Synkronizer 11 when found on your system.
Previous versions of Synkronizer are not affected.

Administrator Privileges

If you do not have administrative rights, you can not install the software.

Installation for Network License

A network license enables the use of a centralized license file stored on a network folder. The path to that location is stored in the registry and can be specified during installation with a command-line option only, best suited for scripted installs. It can also be set or changed in the application itself. (see Registration)

Folders

All files are located in the installation folder typically
`C:\Program Files\Synkronizer\Synkronizer 11\`

The license file is located in

```
C:\ProgramData\Synkronizer\Synkronizer 11
```

The Network License location is stored in the registry under:

```
KeY   : HKLM\Software\[Wow6432Node\]Synkronizer\Synkronizer 11
Name  : LicensePath (String)
Value : \\server\path\license.xml
```

Wow6432Node is used on 64bit windows with 32bit excel

Cached assemblies are located in a subfolder under

```
C:\Users\UserName\AppData\Local\Assembly\dl3
```

User.config files are located in a hashed subfolder under

```
C:\Users\Username\AppData\Local\Microsoft_Corporation\
```

Prerequisites

- Microsoft .NET Framework 4.0
- Visual Studio Tools for Office Runtime 2010

The setup program will search for these prerequisites and install or update them as needed.

5.1 Uninstall

Synkronizer 11 is uninstalled as follows:

1. Open the Windows Control Panel.
2. Select "Programs and Features"
3. Select "Synkronizer 11" and click "Uninstall".

5.2 Network Install

For a network installation, the license file must be stored in a shared network folder. All users working with Synkronizer 11 must have (read-)access to this file. All other files may be located in any directory.

The network license

The network license is a single file that contains all the software, hardware and user data. The network license is installed in a shared network folder a single time. Thereafter, the individual user PCs / workstations (which work with Synkronizer) can be added or removed (without reactivating the license).

Preparation & Registration

1. Create a shared folder on a network.

2. Create a text file named `license.xml`

3. Write the following text in the license file:

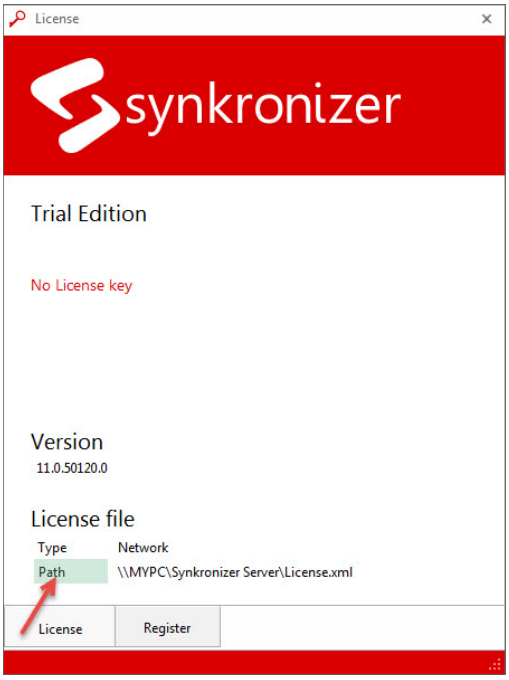
```
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>
<SynkLic />
```

4. Save the file

5. Start Excel.

6. Start Synkronizer and open the License Manager with the following command:
   Project »System » License manager

The following form appears:



Trial Edition

No License key

Version
11.0.50120.0

License file

Type	Network
Path	\\MYPC\Synkronizer Server\License.xml

Click on the path button (see arrow) and choose the network directory.

Note:
The network directory must be an UNC (Uniform Naming Convention) folder which looks as follows:

```
\\ComputerName\SharedFolder\Resource\license.xml
```

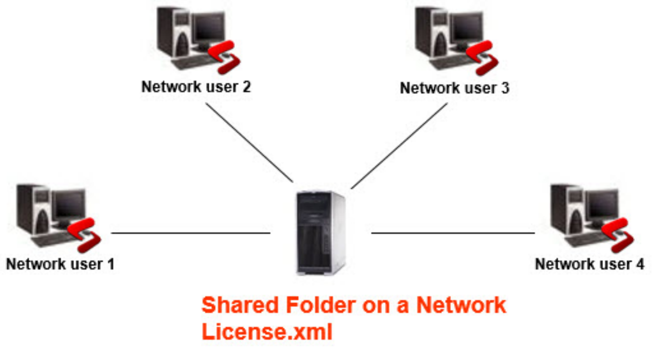
7. Sign up and activate the software

8. Add the network users who work with Synkronizer (see section below).
   Please note that the network directory for the user must be the same as for the administrator.

Installations of network users

The installation of the network users is very easy. The individual network users install Synkronizer normally on their local computer. After that they need to start the license manager and select the License file on the shared folder. If the software is activated and the network users are assigned, the users can immediately begin working with Synkronizer.

Scheme Network



The license file (License.xml) is located on a shared network folder. The individual network users install Synkronizer normally on their local computer and point in the license manager to the license file that is located in the network folder.

Permissions

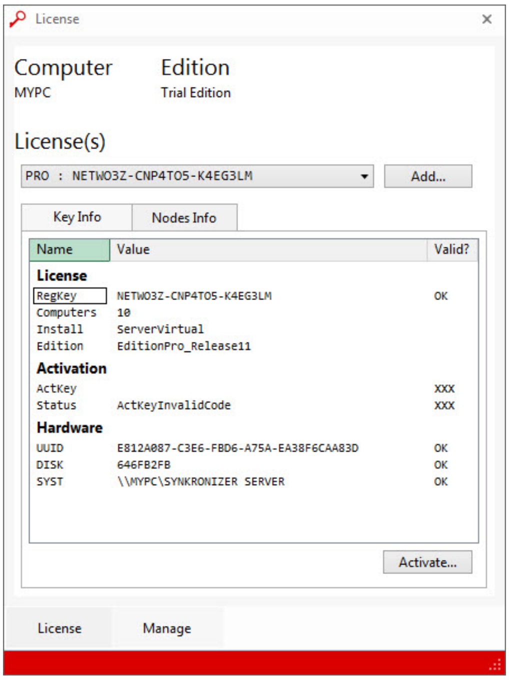
At least one user must have write/modify permission on the license file, while all others only need read permissions. The user(s) with write permission can register/activate the license and manage the list of allowed computers.

5.2.1 Adding nodes

Once the license is activated, you can add the nodes (network users) who will work with Synkronizer. This can be done from any workstation, installed on Synkronizer, as long as the user has write access to the license file.

Add Individual Nodes (workstations/users)

1. Select the Project » System » License Manager.
2. Click on "Manage"
3. Click on the tab "Nodes Info":



Computer
MYPC

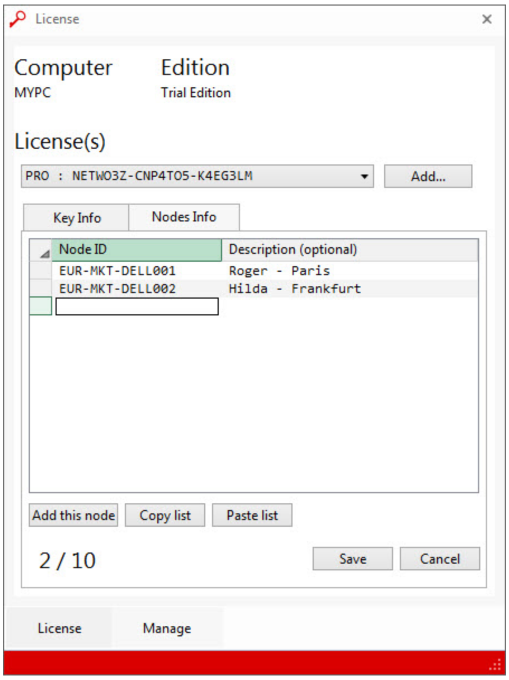
Edition
Trial Edition

License(s)

PRO : NETWO3Z-CNP4TO5-K4EG3LM

Name	Value	Valid?
License		
RegKey	NETWO3Z-CNP4TO5-K4EG3LM	OK
Computers	10	
Install	ServerVirtual	
Edition	EditionPro_Release11	
Activation		
ActKey		XXX
Status	ActKeyInvalidCode	XXX
Hardware		
UUID	E812A087-C3E6-FBD6-A75A-EA38F6CAA83D	OK
DISK	646FB2FB	OK
SYST	\\MYPC\SYNKRONIZER SERVER	OK

4. Now you can add the computer/users, who will work with Synkronizer. Simply enter the correct network user ID and the description in the input fields. If you want to add the computer to which you are working, click on "Add this node":



Computer
MYPC

Edition
Trial Edition

License(s)

PRO : NETWO3Z-CNP4TO5-K4EG3LM

Node ID	Description (optional)
EUR-MKT-DELL001	Roger - Paris
EUR-MKT-DELL002	Hilda - Frankfurt

2 / 10

5. Confirm the mutations with the Save button.

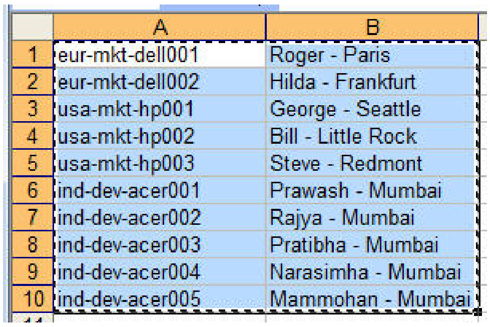
Add a Series of Nodes (workstations/users)

You can prepare a list of workstations (and descriptions) and paste it into the application. The list can be created from a script or typed in a text editor or excel worksheet. Proceed as follows:

1. Start Excel

2. Create a list with the workstation names and users.

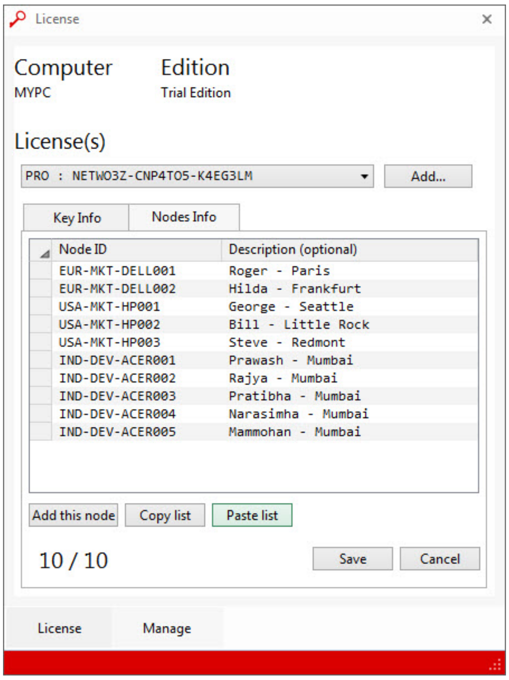
3. Select and copy the list



	A	B
1	eur-mkt-dell001	Roger - Paris
2	eur-mkt-dell002	Hilda - Frankfurt
3	usa-mkt-hp001	George - Seattle
4	usa-mkt-hp002	Bill - Little Rock
5	usa-mkt-hp003	Steve - Redmont
6	ind-dev-acer001	Prawash - Mumbai
7	ind-dev-acer002	Rajya - Mumbai
8	ind-dev-acer003	Pratibha - Mumbai
9	ind-dev-acer004	Narasimha - Mumbai
10	ind-dev-acer005	Mammohan - Mumbai

4. Open the "Manage" tab of Synkronizer

5. Press "Paste list". The list is filled (with a maximum number of licensed computers)



Computer
MYPC

Edition
Trial Edition

License(s)

PRO : NETWO3Z-CNP4TO5-K4EG3LM

Node ID	Description (optional)
EUR-MKT-DELL001	Roger - Paris
EUR-MKT-DELL002	Hilda - Frankfurt
USA-MKT-HP001	George - Seattle
USA-MKT-HP002	Bill - Little Rock
USA-MKT-HP003	Steve - Redmont
IND-DEV-ACER001	Prawash - Mumbai
IND-DEV-ACER002	Rajya - Mumbai
IND-DEV-ACER003	Pratibha - Mumbai
IND-DEV-ACER004	Narasimha - Mumbai
IND-DEV-ACER005	Mammohan - Mumbai

10 / 10

5.3 Scripted Installation

The setup executable is a so-called bootstrapper for the windows installer files contained within, it also takes care of the prerequisites (.NET framework and VSTO runtime)

Command line arguments:

/? or /help
displays help dialog

/extract:<directory>
extracts all files in <directory>

/listlangs
lists languages supported by this setup

/exenoui /qn
launches the EXE setup without UI. Note: run as administrator!

/exebasicui
launches the EXE setup with basic UI

/exelang <langId>
launches the EXE setup with the specified language

/username
username used by the proxy

/password
password used by the proxy

/exelog<path_to_log_file>
creates a log file at specified path

/exenoupdates
does not check for a newer version

/passive licensepath=<Server Path + License.xml>
used for network installation; defines the path and name of the license file

<msiOptions>
options for msiexec.exe running the MSI package

Example:

Standard install without displaying user interface (UI):
Synkronizer.11.0.xxxx.exe /exenoui

Part VI

6 Registration

Upon purchase you will receive a "registration key" for the edition you bought. This registration key is used to "register" and "activate" the software. The activation process is fully automated, it only needs an internet connection. In case firewall settings prevent the program's direct communication with our server, this process can also be performed via website or email.

What happens during the registration process?

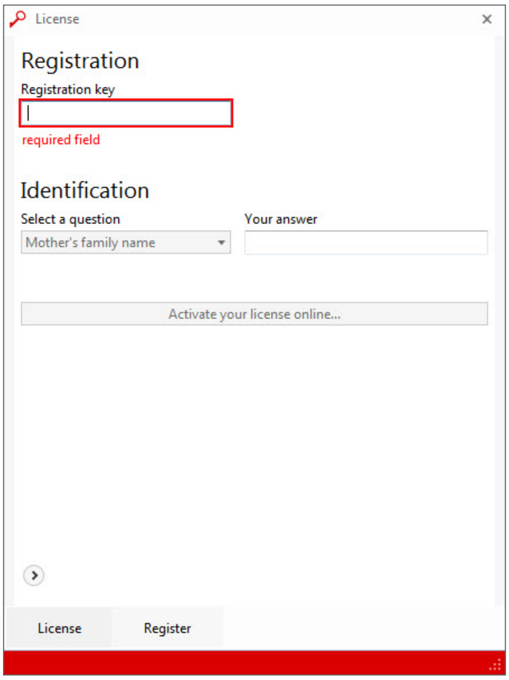
The registration key and some hardware codes are sent to our activation server, which returns one or two activation codes. The activation codes received from the server should "match" the codes sent and will then be stored in the license file. The license file is now uniquely coupled to your PC and your software is activated.

Registration process:

1. Start Synkronizer.

2. Open the license manager with the following command:
   Project » System » License Manager

3. Select the tab "Register":



4. Enter the registration key.

In the case of a multi-user license, you must additionally enter the sequence number of the license (next to the registry key field).

5. Please enter then the secret question.
   Note:
   The security answer is an additional safeguard so that no one else can use your license. If you want to install your license on a second PC, you need to enter the same answer. The secret answer must be at least 5 characters long. The following characters are allowed: A-Z and 0-9.

6. Click "Activate your license online...".
   The registration data is now sent to the activation server and the software is activated.
   If you have no internet connection, you'll need to activate manually. Please read chapter "**Manual Activation**".

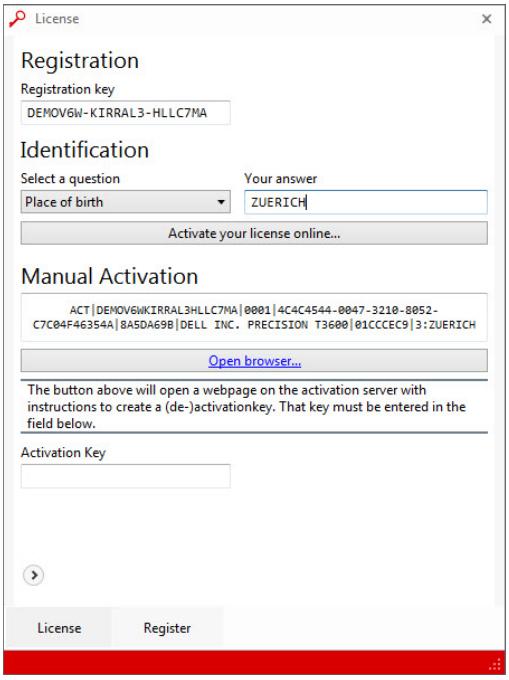
If you have any problems with the registration/activation process, please read chapter "**Problems with registration**".

6.1 Manual Activation

This chapter explains how you can manually activate Synkronizer.

The registrations steps 1 to 6 are explained in the chapter **Registration**.

7. If no Internet connection could be established, the following screen is displayed:



License

Registration

Registration key: DEMOV6W-KIRRAL3-HLLC7MA

Identification

Select a question: Place of birth

Your answer: ZUERICH

Manual Activation

```
ACT|DEMOV6WKIRRAL3HLLC7MA|0001|4C4C4544-0047-3210-8052-
C7C04F46354A|8A5DA69B|DELL INC. PRECISION T3600|01CCCEC9|3:ZUERICH
```

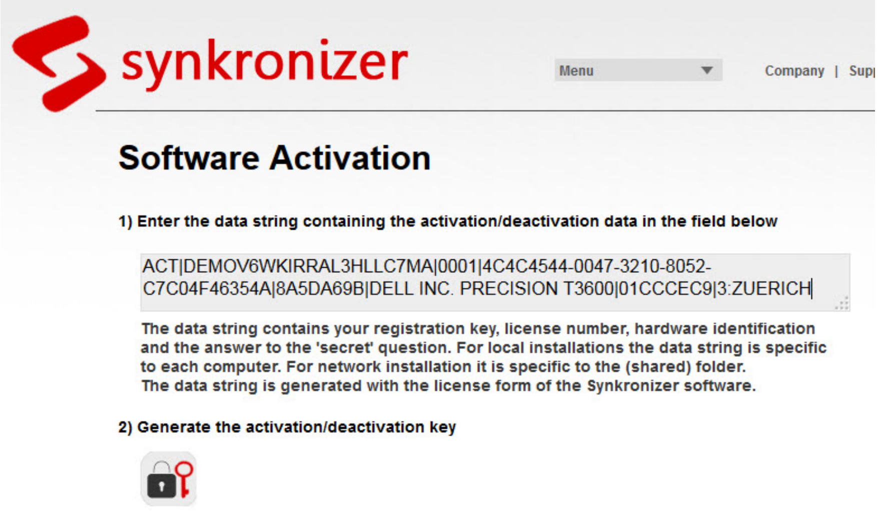
The button above will open a webpage on the activation server with instructions to create a (de-)activationkey. That key must be entered in the field below.

Activation Key

A data string that contains the activation data specific to your hardware is displayed and copied to your clipboard.

8. Click on "Open browser..." or the following link:
   http://www.synkronizer.com/generate-activation-key

   The following page appears:



Software Activation

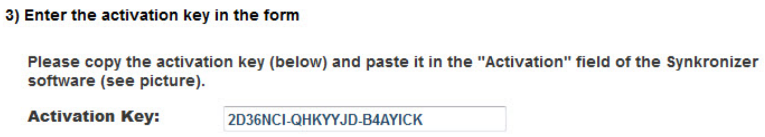
1) Enter the data string containing the activation/deactivation data in the field below

ACT|DEMOV6WKIRRAL3HLLC7MA|0001|4C4C4544-0047-3210-8052-C7C04F46354A|8A5DA69B|DELL INC. PRECISION T3600|01CCCEC9|3:ZUERICH

The data string contains your registration key, license number, hardware identification and the answer to the 'secret' question. For local installations the data string is specific to each computer. For network installation it is specific to the (shared) folder.
The data string is generated with the license form of the Synkronizer software.

2) Generate the activation/deactivation key

9. Paste the datastring into the input field 1) and click on the activate symbol:



3) Enter the activation key in the form

Please copy the activation key (below) and paste it in the "Activation" field of the Synkronizer software (see picture).

Activation Key: 2D36NCI-QHKYYJD-B4AYICK

10. Note the activation key and return to the license manager of Synkronizer:

11. Enter the activation key in the input field "Activation Key".

12. The software is now activated.

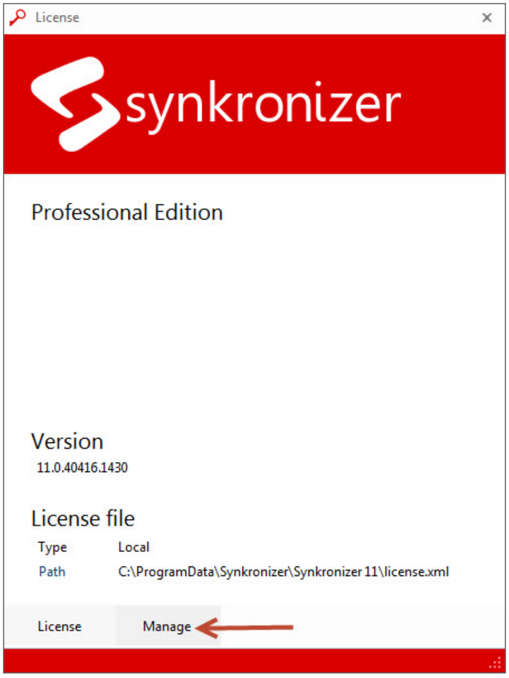
If you have any problems with the registration/activation process please read chapter "**Problems with registration**".

6.2 Transfer License

You can transfer your license to another computer at any time. Simply deactivate the license on the old computer and activate it on the new one.

1. Start Synkronizer.

2. Select Project » Settings » License Manager.



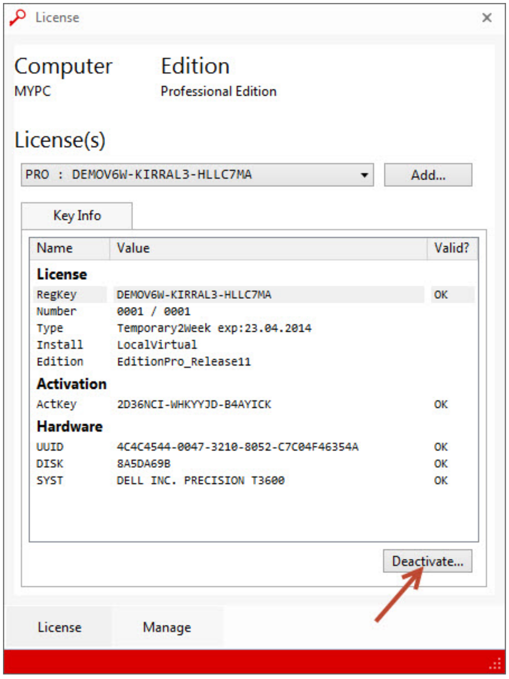
Professional Edition

Version
11.0.40416.1430

License file

Type	Local
Path	C:\ProgramData\Synkronizer\Synkronizer 11\license.xml

3. Select the tab "Manage".



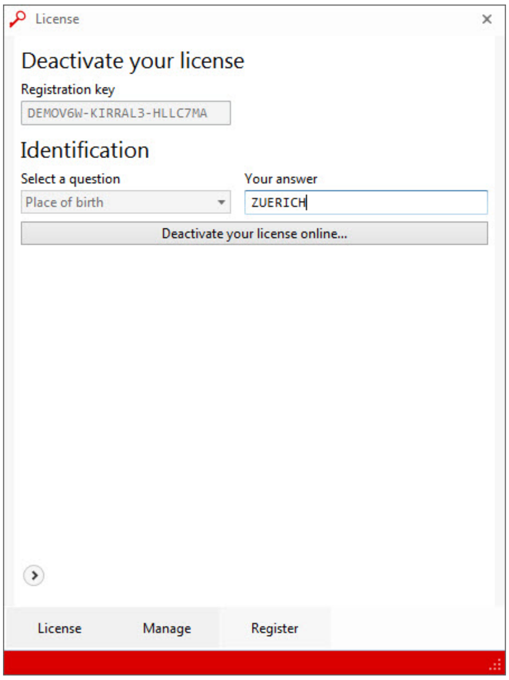
Computer
MYPC

Edition
Professional Edition

License(s)
PRO : DEMOV6W-KIRRAL3-HLLC7MA

Name	Value	Valid?
License		
RegKey	DEMOV6W-KIRRAL3-HLLC7MA	OK
Number	0001 / 0001	
Type	Temporary2Week exp:23.04.2014	
Install	LocalVirtual	
Edition	EditionPro_Release11	
Activation		
ActKey	2D36NCI-WHKYYJD-B4AYICK	OK
Hardware		
UUID	4C4C4544-0047-3210-8052-C7C04F46354A	OK
DISK	8A5DA69B	OK
SYST	DELL INC. PRECISION T3600	OK

4. Click "Deactivate".



License

Deactivate your license

Registration key: DEMOV6W-KIRRAL3-HLLC7MA

Identification

Select a question: Place of birth

Your answer: ZUERICH

5. The register form is displayed. Enter your secret answer and click "Deactivate your license online".

6. The software is deactivated.

If you have no internet connection, you'll need to deactivate the software manually. A manual deactivation works exactly the same like the **manual activation**.
The only difference is, that you'll get an unlock key instead of an activation key. Read this chapter for manually deactivating the software.

Part VII

7 Problems / Errors

Synkronizer 11 is installed but does not appear on the toolbar

The most likely cause is that Synkronizer ended up under "Disabled Add-ins". This may occur when Excel is shut down or crashes while Synkronizer is busy.

Proceed as follows to re-enable Synkronizer:

1. Click the Microsoft Office Button image, click Excel Options, and then click Add-Ins.

2. Check if Synkronizer 11 is listed under "Disabled Application Add-ins" (it should be listed under "Active Application Add-ins").

3. Select & remove Synkronizer from the disabled category. It will then appear either in "Active Application Add-ins" or "Inactive Application Add-ins". If it is listed under "inactive" then go to the ComAdd-in dropdown and activate it.

Now the Synkronizer logo should be visible in the toolbar. If not, proceed as follows:

1. Open the Windows Explorer and select the following directory:
   C:\Program Files\Synkronizer\Synkronizer 11

2. Double-click the file **AddinLoad.bat**.

3. Select **1) Load in Current User (HKCU)**.

4. Close the window.

Error RunMacro:Synkronizer.11.xlam not loaded

This error indicates that you have too rigorous restrictions in the **Trusted Center** of Excel. Proceed as follows to solve this:

1. Verify that you have installed the latest build of Synkronizer 11.

2. Open Excel and select **File » Options » Trust Center » Trust Center Settings**...

3. Select tab "**Trusted Publishers**"

4. Verify that "**XL Consulting GmbH**" is listed in the dialog box.

5. Select tab "**Add-ins**"

6. Verify that no add-ins are disabled.

7. Select tab "**File Block Settings**"

8. Verify that the "**Excel 2007 and later Add-in Files**" can be opened.

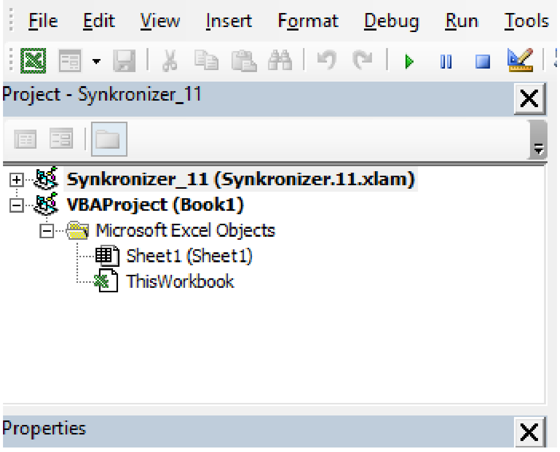
9. Re-start Excel

Synkronizer.11.xlam error: check if references are loaded

1. Open Excel

2. Choose File » Open/Browse
   C:\Program Files\Synkronizer\Synkronizer 11\Synkronizer.11.xlam
   (the xlam is an Add-in file so you will not see it as a workbook)

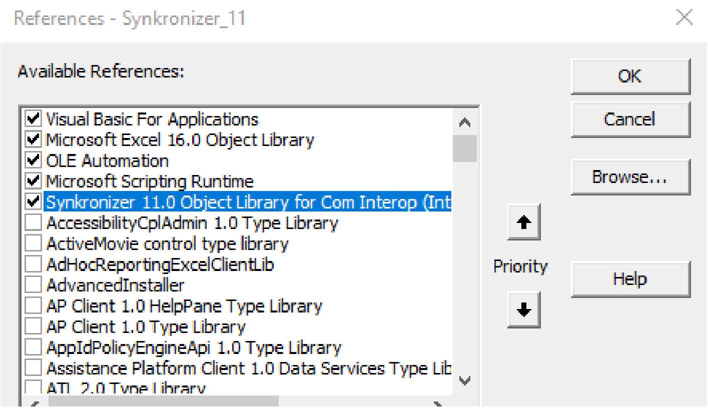
3. Press **[Alt][F11]** (or if your developer tab is visible then do developer/visual basic)
   » the Microsoft Visual Basic Window should open



4. Click on the [+] of the file Synkronizer_11 to open the project for synkronizer_11

5. Enter the password: geheim
   » the treeview with excel objects and modules should now be visible

6. Choose Tools » References it should look like this with five checked references.



References - Synkronizer_11

Available References:

☑ Visual Basic For Applications
☑ Microsoft Excel 16.0 Object Library
☑ OLE Automation
☑ Microsoft Scripting Runtime
☑ Synkronizer 11.0 Object Library for Com Interop (Int
□ AccessibilityCplAdmin 1.0 Type Library
□ ActiveMovie control type library
□ AdHocReportingExcelClientLib
□ AdvancedInstaller
□ AP Client 1.0 HelpPane Type Library
□ AP Client 1.0 Type Library
□ AppIdPolicyEngineApi 1.0 Type Library
□ Assistance Platform Client 1.0 Data Services Type Lib
□ ATL 2.0 Type Library

If any of the first four items is listed as [MISSING] (presumably Scripting Runtime) contact your system administrator.
If Synkronizer 11.0 Object Library for Com Interop is missing than setup failed to do what is should do. Please contact support@synkronizer.com.

7.1 Support

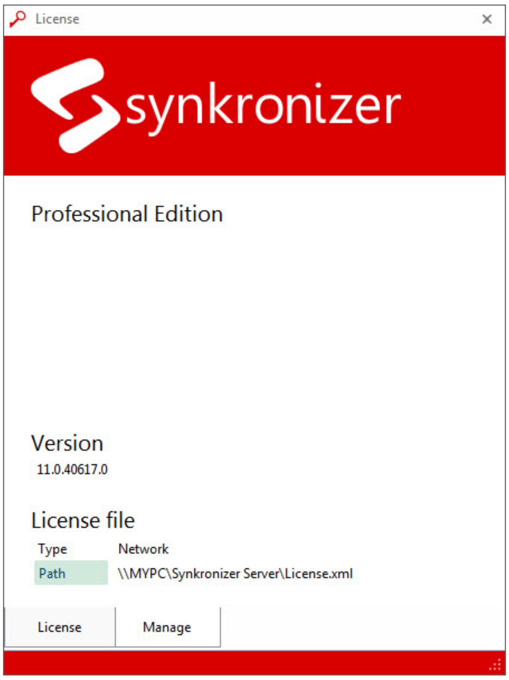
If you should have problems with the software (error messages, unable to activate, etc.), please proceed as follows:

1. Open a new e-mail without entering any content

2. Start the Synkronizer application

3. Select Project » System » License manager
   The license manager is opened

4. Create a screenshot of the license screen:



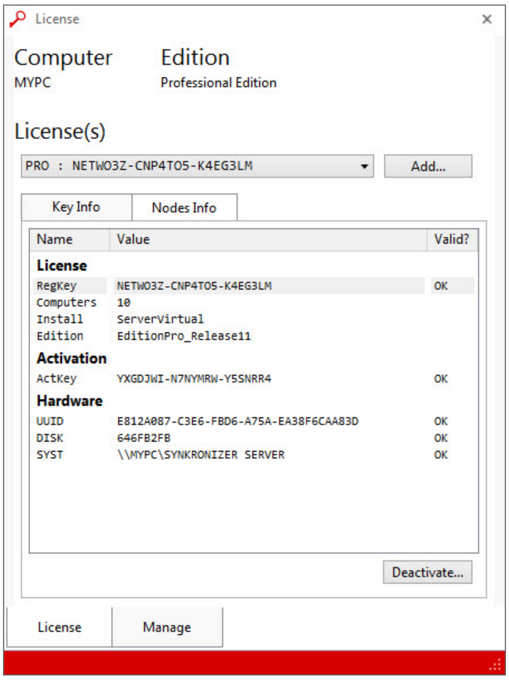
Professional Edition

Version
11.0.40617.0

License file

Type	Network
Path	\\MYPC\Synkronizer Server\License.xml

5. Select the "Manage" tab and create a screenshot of it:



Computer
MYPC

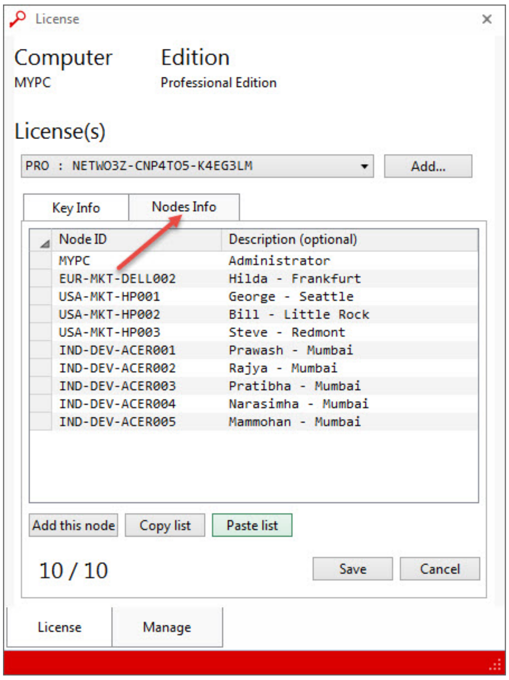
Edition
Professional Edition

License(s)

PRO : NETWO3Z-CNP4TO5-K4EG3LM

Name	Value	Valid?
License		
RegKey	NETWO3Z-CNP4TO5-K4EG3LM	OK
Computers	10	
Install	ServerVirtual	
Edition	EditionPro_Release11	
Activation		
ActKey	YXGDJWI-N7NYMRW-Y5SNRR4	OK
Hardware		
UUID	E812A087-C3E6-FBD6-A75A-EA38F6CAA83D	OK
DISK	646FB2FB	OK
SYST	\\MYPC\SYNKRONIZER SERVER	OK

6. If you are working with a **network license**, click on the "Nodes Info" tab and make a screenshot of it:



Computer
MYPC

Edition
Professional Edition

License(s)
PRO : NETWO3Z-CNP4TO5-K4EG3LM

Node ID	Description (optional)
MYPC	Administrator
EUR-MKT-DELL002	Hilda - Frankfurt
USA-MKT-HP001	George - Seattle
USA-MKT-HP002	Bill - Little Rock
USA-MKT-HP003	Steve - Redmont
IND-DEV-ACER001	Prawash - Mumbai
IND-DEV-ACER002	Rajya - Mumbai
IND-DEV-ACER003	Pratibha - Mumbai
IND-DEV-ACER004	Narasimha - Mumbai
IND-DEV-ACER005	Mammohan - Mumbai

10 / 10

7. Attach the file **license.xml** to the e-mail. The license file is stored in the following folder:

   Local license:
   C:\ProgramData\Synkronizer\Synkronizer 11

   Network license
   Shared network folder

8. Check if a error file was generated in the Windows temp folder. Open the Windows Explorer end enter the folder **%TEMP%**. If a file named **synkronizerError.txt** is there, attach it to the e-mail.

9. Send the screenshots and files to support@synkronizer.com along with any further comments or observations you may have made. Please include information about your environment (e.g., network, Citrix server, terminal server, subnets, etc.) that might help us to evaluate and resolve the problem.

   During business days, you will get an answer from our support team within 24 hours.

7.2 VBA missing

Synkronizer needs VBA macros to run some functions, but sometimes VBA (Visual Basic for Applications) is disabled or not installed.

Check if VBA is available

In Excel a simple check to see if VBA is available is to press **[Alt][F11]**. Normally this command opens a VB Editor window, but when it does not and you hear a beep then VBA is not available. If the Developer Tab is visible in the ribbon you'll notice the VB and Macro buttons in the Code group is disabled.

The most common way System administrators disable VBA is adding a setting in the registry. for more information see:

https://docs.microsoft.com/en-us/previous-versions/office/troubleshoot/office-developer/turn-off-visual-basic-for-application

VBA is disabled

You will find one or more entries called **VBAOFF** under one of the following keys, where XX is the version number of your Excel.

HKEY_CURRENT_USER\Software\Microsoft\Office\XX.0\Common
HKEY_CURRENT_USER\Software\Policies\Microsoft\Office\XX.0\Common
HKEY_LOCAL_MACHINE\Software\Microsoft\Office\XX.0\Common
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Office\XX.0\Common
HKEY_LOCAL_MACHINE\Software\WOW6432Node\Microsoft\Office\XX.0\Common
HKEY_LOCAL_MACHINE\Software\WOW6432Node\Policies\Microsoft\Office\XX.0\Common

The entry is a **REG_DWORD** value named **VBAOFF**. If the value is 1 VBA will be disabled.
To enable VBA the entry must be deleted or the value must be changed to 0.

VBA is not installed

In versions prior to Excel 365 or 2019 VBA was an optional component during setup. In the latest versions it is always installed.

You should find an entry called **vbe7dllpath** under one of the following keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\VBA
HKEY_LOCAL_MACHINE\Software\WOW6432Node\Microsoft\VBA
HKEY_LOCAL_MACHINE\Software\Microsoft\Office\ClickToRun\REGISTRY\MACHINE\Software\Microsoft\VBA

If not then VBA is not properly installed. Proceed as follows:
1. Close Excel
2. Goto Control Panel/Programs & Features or (Apps & Features in latest Win10 installs)

3. Search for your Microsoft Office entry...
4. Click Modify
5. Click Add or Remove Features
6. In the tree find "Office Shared Features", in that branch you'll find "Visual Basic for Applications" . Click the icon and select "Run from my computer".
7. Click Continue and allow setup to process your changes.

Part VIII

8 Developer Edition

8.1 Visual Basic for Applications (VBA)

With the **Developer edition** of Synkronizer, you are able to create **recurring, standardized** and/or **complex comparisons**. To access the automation functionality of Synkronizer, you need to have knowledge about **Visual Basic for Applications (VBA)**. With VBA, you are able to you create your own powerful comparison macros.

We have provided some simple **VBA Examples** and a **"Wrapper" Procedure** to simplify you work.

To test the examples, download the example file **synk11vba.zip** from our server, unzip it and copy the files to a folder of preference.

You can also manually test the examples by using the following steps:

1. Ensure that **Synkronizer 11 Developer** is installed.
2. Start Excel
3. Open a (new) workbook
4. Open the VB Editor (keyboard shortcut Alt-F11)
5. Select your workbook in the Project Explorer window
6. Select menu Tools/References, scroll to **'Synkronizer 11.0 Object Library'**, check it, and press ok.
7. Select menu Insert/Module
8. Copy the **Helper Functions** into this module.
9. Select menu Insert/Module
10. Copy the **VBA Examples** into this module.

The **Object Model Reference** of Synkronizer 11 is described in the following chapters.

8.1.1 Object Model Reference

Object Model Reference of Synkronizer 11

The object model of Synkronizer 11 looks as follows:

Excel
- **ComAddins("Synkronizer.AddIn")**
  - **Object = Synkronizer.Application**
    - **Project**
      - **Files**
      - **Pairs**
        - **Pair**

    - Results
- Settings
- Results

8.1.1.1 Application Object

Top level object. Entry point for all automation.

Methods

Name	Description
NewProject	Returns a project with empty files, default settings and manual matching
OpenProject	Creates a new project with empty files, default settings and manual matching.

Properties

Name	Description
ActiveProject	Project that is currently active.
DisplayUI	Determines if user interface is displayed or not.

Remarks

The application object is only accessible when you have a licensed Developer Edition

Example

To access the Synkronizer Application, you should do the following:.

```
Dim cai As COMAddIn
Dim snk As Synkronizer.Application

Set cai = Application.COMAddIns("Synkronizer.Addin")
If Not cai.Connect Then
  cai.Connect = True
End If
Set snk = cai.Object
```

8.1.1.1.1 Methods

8.1.1.1.1.1 NewProject

Creates a new project with empty files, default settings and manual matching.

Syntax

expression.**NewProject**

expression. A variable representing an Application object.

Parameters

none

Return Value

none

Remarks

If another project is active, it will be closed first.

8.1.1.1.1.2 OpenProject

Opens a project as defined in specified xml file.

Syntax

expression.**OpenProject*(file name)***

expression. A variable representing an **Application** object.

Parameters

Name	Type	Description
file name	String	Path and file name of the project file. (File extension is '.xml').

Return Value

none

Remarks

If another project is active it will be closed first.

8.1.1.1.2 Properties

8.1.1.1.2.1 ActiveProject

Returns the project that is currently active. **Read only** .

Syntax

expression.**ActiveProject**

expression. A variable representing an **Application** object.

8.1.1.1.2.2 DisplayUI

Shows or hides the Synkronizer user interface.

Syntax

expression.**DisplayUI**
expression. A variable representing a **Application** object.

8.1.1.2 Project Object

The project is the central object for working with Synkronizer.

Methods

Name	Description
Close	Closes the current project. Optionally closes the source, target and (if available) difference report files
Execute	Executes the compare process.
Save	Saves the project's configuration to the specified path.

Properties

Name	Description
Files	Container for the Files (workbooks) of the project.
IsDisposed	Project was deactivated and internal object set to nothing.
IsValid	Checks if files are valid and pairs exist and are all valid.
Pairs	Container for the Pairs (worksheets) of the project.
ReportWorkbook	Report workbook (when created).
Results	Container for the (combined) Results of all Pairs.
Settings	Container for all Settings of the project.

Events

Name	Description
Progress	Raised during comparison execution to indicate running subprocess.

8.1.1.2.1 Methods

8.1.1.2.1.1 Close

Closes the project results. Optionally closes the workbooks (without saving!)

Syntax

expression.**Close*(CloseFiles,DisplayUndo)***
expression. A variable representing a **Project** object.

Parameters

Name	Type	Description
CloseFiles	Boolean	Closes the compared files (workbooks).
DisplayUndo	Boolean	Optional. If true and highlighting or outline are activated, the user

		will be given the option to undo

Return Value

none

Remarks

If you want the files to be saved (with possible highlighting and outline), you must do so prior to calling the Close method.

8.1.1.2.1.2 Execute

Executes the compare process.

Syntax

expression.**Execute**

expression. A variable representing a **Project** object.

8.1.1.2.1.3 Save

Saves the project's configuration to the specified path.

Syntax

expression.**Save*(FilePath)***

expression. A variable representing a **Project** object.

Parameters

Name	Type	Description
FilePath	String	Path and file name of the project file. (File extension is '.xml').

Return Value

none

8.1.1.2.2 Properties

8.1.1.2.2.1 Files

Container for the Files (workbooks) of the project. **Read only** **Files** object.

Syntax

expression.**Files**

expression. A variable representing a **Project** object.

8.1.1.2.2.2 IsDisposed

This property checks if the project was deactivated and internal object set to nothing. **Read only Boolean** property,

Syntax

expression.***IsDisposed***

expression. A variable representing a **Project** object.

8.1.1.2.2.3 IsValid

Checks if files are valid and pairs exist and are all valid. **Read only Boolean** property,

Syntax

expression.***IsValid***

expression. A variable representing a **Project** object.

8.1.1.2.2.4 Pairs

Container for the pairs (worksheets) of the project. **Read only Pairs** object.

Syntax

expression.**Pairs**

expression. A variable representing a **Project** object.

8.1.1.2.2.5 ReportWorkbook

Report workbook (when created). **Read only Workbook** object.

Syntax

expression.**ReportWorkbook**

expression. A variable representing a **Project** object.

8.1.1.2.2.6 Results

Container for the (combined) Results of all pairs. **Read only Results** object.

Syntax

expression.**Results**

expression. A variable representing a **Project** object.

8.1.1.2.2.7 Settings

Container for all settings of the project. **Read only** **Settings** object.

Syntax

expression.**Settings**

expression. A variable representing a **Project** object.

8.1.1.2.3 Events

8.1.1.2.3.1 Progress

Raised during compare execution to indicate running subprocess.

Remarks

You can use this event to display a progress indicator.

8.1.1.3 Files Object

Container for the two files (workbooks) to compare.

Methods

Name	Description
Load	Loads the two files from disk.
Save	Saves the source file and the target file.

Properties

Name	Description
file name	Returns the full path and file name of either of the files.
IsValid	Returns true if both files are loaded and can be compared.
Window	Returns the window object of either of the files.
Workbook	Returns the workbook object of either of the files.

Events

Name	Description

none

8.1.1.3.1 Methods

8.1.1.3.1.1 Load

Loads the two files from disk.

Syntax

expression.**Load*(file name0,file name1)*** expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
file name0	String	Full path for 1st file (also referred to as source).
file name1	String	Full path for 2nd file (also referred to as target).
NoArrange	Boolean	Optional. Set true if workbooks don't need to be arranged horizontally or vertically.

Return Value

none

8.1.1.3.1.2 Save

Loads the two files from disk.

Syntax

expression.**Save*(Save0,Save1,Path0,Path1)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
Save0	Boolean	Set true if the source file needs to be saved.
Save1	Boolean	Set true if the target file needs to be saved.
Path0	String	Optional. Specifies the path if the source file needs to be saved in a different location.
Path1	String	Optional. Specifies the path if the target file needs to be saved in a different location.

Return Value

none

8.1.1.3.2 Properties

8.1.1.3.2.1 FileName

Returns the full path and file name of either of the Files. **Read only String** value.

Syntax

expression.**file name*(id)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	sideID	Selects the source or target file.

8.1.1.3.2.2 IsValid

Returns true if both files are loaded and can be compared. **Read only Boolean** value.

Syntax

expression.**IsValid**

expression. A variable representing a **Files** object.

8.1.1.3.2.3 Window

Returns the window object of either of the files. **Read only Window** object.

Syntax

expression.**Window*(id)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	sideID	Selects source or target file.

8.1.1.3.2.4 Workbook

Returns the workbook object of either of the files. **Read only Workbook** object.

Syntax

expression.**Workbook*(id)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	sideID	Selects source or target file.

8.1.1.4 Pairs Object

Collection of pair objects.

Methods

Name	Description
AddMatched	Adds matched pairs using MatchType and MatchInclude properties.
AddPair	Adds a pair to the Pairs collection.
Clear	Clears the pairs collection.
Remove	Removes specified pair from the collection.

Properties

Name	Description
Count	Returns the number of members in the collection.
IsValid	Returns true if pairs exist and each pair is valid.
Item	Returns a single item (pair) from the collection.
MatchInclude	Returns or sets options specifying types of worksheets to include by **AddMatched** method.
MatchType	Returns or sets a constant specifying how worksheets will be matched by **AddMatched** method.
PasswordList	Returns or sets a semicolon delimited list of passwords needed to access protected sheets.

Events

Name	Description

none

Remarks

You can add pairs individually with the **Add** Method or use the **AddMatched** method to automatically add pairs for all worksheets that have matching names or indexes.

You can retrieve a specific pair with Pairs(Index) or Pairs.Item(Index).
You can remove a specific pair with Pairs.Remove(Index).
You can remove all pairs by using the Clear Method.

8.1.1.4.1 Methods

8.1.1.4.1.1 AddMatched

Adds matched pairs using MatchType and MatchInclude properties.

Syntax

*expression.***AddMatched**
expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with the same name. Protected and hidden sheets are included.

```
With .Pairs
   'include protected and hidden sheets
   .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
   'match all sheets with same name
   .MatchType = MatchType_AllByName
   'add sheets
   .AddMatched
End With
```

8.1.1.4.1.2 AddPair

Adds a pair to the Pairs collection.

Syntax

expression.**AddPair*(Sheet0,Sheet1,Range0,Range1,DBoptions,WSoptions,DBrow,DBkeys)***
expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
Sheet0	**String**	A worksheet name of the source file.
Sheet1	**String**	A worksheet name of the target file.
Range0	**String**	A range address of the source worksheet.
Range1	**String**	A range address of the target worksheet.
DBOptions	**DBOptionFlag**	If the worksheet is a database, enter database options here.
WSoptions	**WSOptionFlag**	If the worksheet should be linked 1 on 1, enter linking options.
DBRow	**Long**	If database comparison, enter row with database column/ field names.
DBKeys	**String**	If database comparison, enter the column number(s) which contain(s) the primary key(s). Separate numbers by semicolon(s).

Return Value

Returns a **Pair** object

8.1.1.4.1.3 Clear

Clears the pairs collection.

Syntax

expression.**Clear**
expression. A variable representing a **Pairs** object.

Return Value

none

8.1.1.4.1.4 Remove

Removes specified pair from the collection.

Syntax

expression.**Remove*(index)***

expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
index	Long	A valid ID or index for the pair.

Return Value

none

8.1.1.4.2 Properties

8.1.1.4.2.1 Count

Returns the number of members in the collection. **Read only long** value.

Syntax

expression.**Count**

expression. A variable representing a **Pairs** object.

8.1.1.4.2.2 IsValid

Returns true if pairs exist and each pair is valid. **Read only Boolean** value.

Syntax

expression.**IsValid**

expression. A variable representing a **Pairs** object.

8.1.1.4.2.3 Item

Returns a single item (Pair) from the collection. **Pair** object.

Syntax

expression.**Item*(index)***

expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
index	Long	A valid ID or index for the pair.

8.1.1.4.2.4 MatchInclude

Returns or sets options specifying types of worksheets to include by **AddMatched** method. **MatchIncludeFlag** constant.

Syntax

*expression.***MatchInclude**

expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with the same name. Protected and hidden sheets are included.

```
With .Pairs
   'include protected and hidden sheets
   .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
   'match all sheets with same name
   .MatchType = MatchType_AllByName
   'add sheets
   .AddMatched
End With
```

8.1.1.4.2.5 MatchType

Returns or sets a variable specifying how worksheets will be matched by **AddMatched** method. **MatchType** constant.

Syntax

*expression.***MatchType**

expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with equal names. Protected and hidden sheets are included.

```
With .Pairs
   'include protected and hidden sheets
   .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
   'match all sheets with same name
   .MatchType = MatchType_AllByName
   'add sheets
   .AddMatched
```

```
End With
```

8.1.1.4.2.6 PasswordList

Returns or sets a semicolon delimited list of passwords needed to access protected sheets. **String** value.

Syntax

expression.**PasswordList**

expression. A variable representing a **Pairs** object.

Example

This example adds two password protected worksheet pairs.

```
With .Pairs
   'add worksheets
   .AddPair "Customer List", "Customer List"
   .AddPair "Customer Profile", "Customer Profile"

   'enter passwords (separated by semicolons)
   .PasswordList = "abc;def"
End With
```

8.1.1.5 Pair Object

Container for a pair of worksheets to compare.

Methods

Name	Description
Activate	Activates the pair object.

Properties

Name	Description
DBKeys	Returns or sets a semicolon delimited list of column indices used as primary key for database.
DBOptions	Returns or sets options for sorting or grouping database records.
DBRow	Returns or sets the index for the row which contains the headers / field names.
IsValid	Returns true if the pair is valid.
Range	Returns the range object for the specified member of the pair.
RangeAddr	Returns or sets an optional range address for the specified member of the pair.
Results	Returns the number of differences found.
Sheet	Returns the worksheet object for the specified member of the pair.

SheetName	Returns the name of the worksheet for the specified member of the pair.
WSOptions	Returns the row/column linking options.

Remarks

Specify a Range

Normally, the entire data range of the sheet is used, but you can limit the range by using the **RangeAddr** property.

Compare as Database

You can compare the sheets as databases if
· your data is organized in a tabular layout, with descriptive headers above each column.
· rows can be uniquely identified by a primary key.

You must set the primary key using the **DBKeys** property.
You can specify the starting row of the database with the **DBRow** property.
If you want the records sorted or the results grouped, use **DBOptions** .

8.1.1.5.1 Methods

8.1.1.5.1.1 Activate

Activates the pair.

Syntax

expression.**Activate**

expression. A variable representing a **Pair** object.

8.1.1.5.2 Properties

8.1.1.5.2.1 DBKeys

Returns or sets a semicolon delimited list of column indices used as primary key for database. **String** value.

Syntax

expression.**DBKeys**

expression. A variable representing a **Pair** object.

Example

This example groups the records.

```
With oProj.Pairs.Item(0)
   'row number of database heading
   .DBRow = 3
   'create primary key; column no. of source file
   .DBKeys = "2;3"
```

```
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.2 DBOptions

Returns or sets options for sorting or grouping database records. **DBOptionFlag** value.

Syntax

*expression.***DBOptions**

expression. A variable representing a **Pair** object.

Example

This example groups the records.

```
With oProj.Pairs.Item(0)
    'row number of database heading
    .DBRow = 3
    'create primary key; column no. of source file
    .DBKeys = "2;3"
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.3 DBRow

Returns or sets the index for the row which contains the headers. **Long** value.

Syntax

*expression.***DBRow**

expression. A variable representing a **Pair** object.

Example

This example defines the 3rd row as database heading.

```
With oProj.Pairs.Item(0)
    'row number of database heading
    .DBRow = 3
    'create primary key; column no. of source file
    .DBKeys = "2;3"
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.4 IsValid

Returns true if the pair is valid. **Read only Boolean** object.

Syntax

expression.***IsValid***

expression. A variable representing a **Pair** object.

8.1.1.5.2.5 Range

Returns the range object for the specified member of the pair. **Read only Range** object.

Syntax

expression.**Range*(id)***

expression. A variable representing a **Pair** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target worksheet.

Remarks

To change the Range use the **RangeAddr** property.

8.1.1.5.2.6 RangeAddr

Returns or sets an optional range address for the specified member of the pair. **Read only String** value.

Syntax

expression.**RangeAddr*(id)***

expression. A variable representing a **Pair** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target worksheet.

Example

This example sets the range address.

```
With .Pairs
   .Add "db1", "db1"
   .Item(1).RangeAddr(sideID_src) = "A1:H50"
   .Item(1).RangeAddr(sideID_tgt) = "A1:H50"
```

```
End With
```

8.1.1.5.2.7 Sheet

Returns the Worksheet object for the specified member of the pair. **Read only Worksheet** object.

Syntax

expression.**Sheet*(id)***

expression. A variable representing a **Pair** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target worksheet.

8.1.1.5.2.8 SheetName

Returns the name of the worksheet for the specified member of the pair. **Read only String** value.

Syntax

expression.**SheetName*(id)***

expression. A variable representing a **Pair** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target worksheet.

8.1.1.5.2.9 WSOptions

Returns or sets the name of the linking options.

Syntax

expression.**WSOptions**

expression. A variable representing a **WSOptions** object.

Example

This example links the rows and columns '1 on 1'.

```
With oProj.Pairs.Item(0)
   .WSOptions = WSOptionFlag_Rows1on1 + WSOptionFlag_Cols1on1
End With
```

8.1.1.6 Settings Object

Container for a group of properties that specify how files will be compared and what actions will be performed.

Methods

Name	Description
Clear	Clears all settings.

Properties

Name	Description
Contents	Return or sets the additional content to be compared.
FilterEquivalents	Returns or sets the equivalent values which will be ignored.
Filters	Returns or sets a series of options specifying which differences to filter (ignore).
FilterTolerance	Returns or sets a number below which numeric differences will be ignored.
Formats	Returns or sets a series of options specifying which formatting properties to compare.
Highlight	Returns or sets a variable specifying which highlighting to perform.
Outline	Returns or sets a series of options specifying which group of rows and columns to show or hide.
Report	Returns or sets a variable specifying which report to create.

Events

Name	Description

none

8.1.1.6.1 Methods

8.1.1.6.1.1 Clear

Clears all settings.

Syntax

expression.**Clear**

expression. A variable representing a **Settings** object.

Return Value

none

8.1.1.6.2 Properties

8.1.1.6.2.1 Contents

Returns or sets a series of options specifying which formatting properties to compare. **Contents** constant.

Syntax

expression.**Contents**

expression. A variable representing a **Settings** object.

Example

This example also compares comments and Excel names.

```
With .Settings
  .Contents = ContentFlag_Comments + ContentFlag_Names
End With
```

8.1.1.6.2.2 FilterEquivalents

Returns or sets the equivalent values. **String** value.

Syntax

expression.**FilterEquivalents**

expression. A variable representing a **Settings** object.

Example

This example compares the worksheets using filters.

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpace
  .FilterTolerance = 0.01
  .FilterEquivalents = "yes,ja;no,nein"
End With
```

8.1.1.6.2.3 Filters

Returns or sets a series of options specifying which differences to filter (ignore). **FiltersFlag** constant.

Syntax

expression.**Filters**

expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example compares the worksheets using filters.

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpace
  .FilterTolerance = 0.01
  .FilterEquivalents = "yes,ja;no,nein"
End With
```

8.1.1.6.2.4 FilterTolerance

Returns or sets a number below which numeric differences will be ignored. **Double** value.

Syntax

expression.**FilterTolerance**

expression. A variable representing a **Settings** object.

Example 1

This example will filter out any results where numeric differences are less than or equal to 0.01 (e.g. 0.01 vs .015):

```
With .Settings
  .Filters = FilterFlag_Enabled
  .FilterTolerance = 0.01
End With
```

Example 2

This example will filter out any results where numeric differences are less than or equal to 1 percent of the sc

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_RelativeTolerance
  .FilterTolerance = 0.01
End With
```

8.1.1.6.2.5 Formats

Returns or sets a series of options specifying which formatting properties to compare. **FormatsFlag** constant.

Syntax

expression.**Formats**

expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example compares font formats.

```
With .Settings
  .Formats = FormatFlag_Enabled + FormatFlag_Font
End With
```

8.1.1.6.2.6 Highlight

Returns or sets a constant specifying which Highlighting to perform. Uses **HighlightType** constants.

Syntax

expression.**Highlight**

expression. A variable representing a **Settings** object.

Example

This example highlights the differences.

```
With .Settings
  .Highlight = HighlightType_Standard
End With
```

8.1.1.6.2.7 Outline

Returns or sets a series of options specifying which group of Rows and Columns to Show or Hide. Uses **OutlineFlag** constants.

Syntax

expression.**Outline**

expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example shows only the differences. Identical rows are hidden.

```
With .Settings
  .Outline = OutlineFlag_Enabled + _
             OutlineFlag_DifferentRows + OutlineFlag_MissingRows + OutlineFlag_Dupl
             OutlineFlag_MissingCols
End With
```

8.1.1.6.2.8 Report

Returns or sets a constant specifying which Report to create. Uses **ReportType** constants.

Syntax

expression.**Report**

expression. A variable representing a **Settings** object.

Example

This example creates a difference report.

```
With .Settings
   .Report = ReportType_Standard
End With
```

8.1.1.7 Results Object

Container for the number of differences found.

Methods

Name	Description
none	

Properties

Name	Description
ArrayCounts	Returns an array with the overall differences found.
ArrayDetails	Returns an array with the individual differences found.
DifferentComments	Returns the number of cells with cell comments.
DifferentContents	Returns the number of cells with different (entered) values and formulas.
DifferentFormats	Returns the number of cells with different formats.
DifferentNames	Returns the number of different Excel names.
DifferentValues	Returns the number of cells with different calculated values (or results).
DuplicateKeys	Returns the number of duplicate keys (database only).
DuplicateRecords	Returns the number of duplicate records (database only).
MissingCols	Returns the number of missing columns (or fields).
MissingRows	Returns the number of missing rows (or records).
MissingSheets	Returns the number of missing worksheets (only for projects with automatching).
Sum	Returns the total number of differences found.
SumByType	Returns the number of specified differences found.
SumText	Returns the results as a formatted text for message box.

SumTextByType	Returns the number of specified differences found as text.

Events

Name	Description
none	

Remarks

- If the results object is a child of the **Project** object, then combined results are returned.
- If the results object is a child of the **Pair** object, then individual results are returned.

8.1.1.7.1 Properties

8.1.1.7.1.1 ArrayCounts

Returns an array with the overall number of differences found as text. **Read only Array**.

Syntax

*expression.***ArrayCounts**
expression. A variable representing a **Results** object.

Description of Array

This property returns an array. Each difference is represented by a single element of the array with 11
columns. The first element of the array contains the column headings.

Column #	Name	Description
0	Source	Name of source worksheet.
1	Total	Total differences of worksheet pair.
2	Columns	Overall differences of missing columns (source¦target).
3	DupRecs	Overall differences of duplicate records.
4	DupKeys	Overall differences of duplicate keys.
5	Rows	Overall differences of missing rows (source¦target).
6	Content	Overall differences of entered values/formulas.
7	Values	Overall differences of calculated values.
8	Formats	Overall differences of format differences.
9	Comments	Overall differences of cell comments.
10	Names	Overall differences of cell Excel names.

8.1.1.7.1.2 ArrayDetails

Returns an array with the detail information of the specified differences found as text. **Read only Array**.

Syntax

*expression.***ArrayDetails*(Type)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return. If not set, all result types are returned.

Description of Array

This property returns an array. Each difference is is represented by a single element of the array with 11
columns. The first element of the array contains the column headings.

Column #	Name	Description
0	Source	Name of source worksheet.
1	Adr1	Range address of source worksheet
2	Adr2	Range address of target worksheet
3	Val1	Cell value/formula of source worksheet
4	Val2	Cell value/formula of target worksheet
5	Key	Primary key (only if database)
6	Fld	Field name (only if database)
7	TypeName	Type of difference. The following difference types are displayed: - MissingCol (different columns) - DuplicateRec (duplicate records) - DuplicateKey (duplicate keys) - MissingRow (different rows) - DifferentContent (different entered values/formulas) - DifferentValue (different calculated formulas/results) - DifferentFormat (different formats) - DifferentComment (different cell comments) - DifferentName (different Excel names)
8	TypeSort	Type number of difference.
9	ColSort	Column number of difference.
10	RowSort	Row number of difference.

8.1.1.7.1.3 DifferentComments

Returns the number of different cell comments. **Read only Long** value.

Syntax

expression.**DifferentComments**

expression. A variable representing a **Results** object.

8.1.1.7.1.4 DifferentContents

Returns the number of cells with different (entered) vales and/or different (entered) formulas. **Read only Long** value.

Syntax

expression.**DifferentContents**

expression. A variable representing a **Results** object.

8.1.1.7.1.5 DifferentFormats

Returns the number of cells with different formats. **Read only Long** value.

Syntax

expression.**DifferentFormats**

expression. A variable representing a **Results** object.

8.1.1.7.1.6 DifferentNames

Returns the number of different Excel names. **Read only Long** value.

Syntax

expression.**DifferentNames**

expression. A variable representing a **Results** object.

8.1.1.7.1.7 DifferentValues

Returns the number of cells with different cell results. **Read only Long** value.

Syntax

expression.**DifferentValues**

expression. A variable representing a **Results** object.

8.1.1.7.1.8 DuplicateKeys

Returns the number of duplicate keys (database only). **Read only Long** value.

Syntax

expression.**DuplicateKeys*(id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.9 DuplicateRecords

Returns the number of duplicate records (database only). **Read only Long** value.

Syntax

expression.**DuplicateRecords*(id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.10 MissingCols

Returns the number of missing columns (or fields). **Read only Long** value.

Syntax

expression.**MissingCols*(id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.11 MissingRows

Returns the number of missing rows (or records). **Read only Long** value.

Syntax

expression.**MissingRows*(id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.12 MissingSheets

Returns the number of missing worksheets (auto matching only). **Read only Long** value.

Syntax

expression.**MissingSheets*(id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.13 Sum

Returns the total number of differences found. **Read only Long** value.

Syntax

expression.**Sum**

expression. A variable representing a **Results** object.

8.1.1.7.1.14 SumByType

Returns the number of specified differences found. **Read only Long** value.

Syntax

expression.**SumByType*(Type,id)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return.
id	**sideID**	Optional. Only needed for MissingCols and MissingRows. Leave blank to return combined results.

Example

This example prints out some differences.

```
With oProj.Results
  'print out entered value/formula differences
  Debug.Print "Different values/formulas: " & .SumByType(ResultType_DifferentConter

  'print out missing rows in source file
  Debug.Print "Missing Rows Source: " & .SumByType(ResultType_MissingRow, sideID_sr

  'print out missing rows in target file
  Debug.Print "Missing Rows Target: " & .SumByType(ResultType_MissingRow, sideID_tg
End With
```

8.1.1.7.1.15 SumText

Returns a formatted text that can be used in a messagebox. **Read only String** value.

Syntax

expression.**SummaryText**

expression. A variable representing a **Results** object.

8.1.1.7.1.16 SumTextByType

Returns the number of specified differences found as text. **Read only String** value.

Syntax

expression.**SumTextByType*(Type)***

expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return.

8.1.1.8 Enumerations

8.1.1.8.1 Content Flag

Constants used to by the **Contents** property in Settings.

Name	Description
ContentFlag_Comments	If this flag is set, the comments will be compared.
ContentFlag_Names	If this flag is set, the Excel names will be compared.
ContentFlag_None	No additional content will be compared.

8.1.1.8.2 DBOptionFlag

Flags for pair's **DBoptions**

Name	Description
DBOptionFlag_AddID	Adds an ID in the last column of the database to aid sorting and filtering.
DBOptionFlag_Group	Records are grouped by result after comparing.
DBOptionFlag_RelaxedKeys	When selecting this option, primary key differences in spaces and upper/lower case are ignored.
DBOptionFlag_Sort	Records are sorted (on fields in primary key) before comparing.

8.1.1.8.3 FilterFlag

Constants used to by the **Filters** property in Settings.

Name	Description
FilterFlag_Constants	All constants are ignored. Only formulas are compared.
FilterFlag_Datatype	Differences in datatype are ignored. Use to ignore differences between date values and string values representing dates.
FilterFlag_Enabled	Enable or disable all other filters options.
FilterFlag_Formulas	All formulas are ignored. Only constants are compared.
FilterFlag_HiddenCols	Data in hidden columns will be ignored.
FilterFlag_HiddenRows	Data in hidden rows will be ignored.
FilterFlag_None	No filters will be used.
FilterFlag_StringCase	Text compare case Insensitive. "ABC" vs "abc" will be considered as equal.
FilterFlag_StringSpace	Text compare trim text. "ABC " vs "ABC" will be considered as equal.
FilterFlag_RelativeTolerance	If this flag is on, then the relative differences (%) are filtered out (instead of numeric differences #)

8.1.1.8.4 FormatFlag

Constants to be used for the **Formats** property in Settings.

Name	Description
FormatFlag_Alignment	Compare cell alignment properties.
FormatFlag_Border	Compare cell border properties.
FormatFlag_Enabled	Enable or disable all other formats options.
FormatFlag_Font	Compare cell font properties.
FormatFlag_None	No formats are compared.
FormatFlag_Number	Compare cell number (numberformat) properties.
FormatFlag_Pattern	Compare cell pattern (interior/fill) properties.
FormatFlag_Protection	Compare cell protection properties.

8.1.1.8.5 HighlightType

Constants to be used for the **Highlight** property in Settings.

Name	Description
HighlightType_None	Differences are not highlighted.
HighlightType_Standard	Differences are highlighted. No Cell coloring for all cells if not removed before processing.
HighlightType_WithReset	Differences are highlighted. Cell coloring for all cells is removed before processing.

8.1.1.8.6 MatchIncludeFlag

Flags specifying a group of sheets to include during Matching.

Name	Description
MatchIncludeFlag_EmptySheets	Empty sheets will be included by **AddMatched**
MatchIncludeFlag_HiddenSheets	Hidden sheets will be included by **AddMatched**
MatchIncludeFlag_None	Visible worksheets with content will be included. Default setting.
MatchIncludeFlag_ProtectedSheets	Protected sheets will be included by **AddMatched**

8.1.1.8.7 MatchType

List of valid types of matching.

Name	Description
MatchType_AllByCodeName	All included sheets are matched by code name.
MatchType_AllByIndex	All included sheets are matched by index.
MatchType_AllByName	All included sheets are matched by name.
MatchType_FirstByCodeName	The first matched sheets with the same code name are added.
MatchType_FirstByIndex	The first matched sheets are added.
MatchType_FirstByName	The first matched sheets with the same name are added.
MatchType_Manual	Pairs were modified or added with **Add**

8.1.1.8.8 OutlineFlag

Constants used to by the **Outline** property in Settings.

Name	Description
OutlineFlag_DifferentRows	Hide different column: (linked, some cells are different).
OutlineFlag_DuplicateRows	Hide rows with either duplicate keys or records (only in database mode).
OutlineFlag_Enabled	Enable or Disable all other Outline options.
OutlineFlag_IdenticalRows	Hide identical rows (linked, without cell differences).
OutlineFlag_MissingCols	Hide missing columns: (not occurring in the other worksheet).
OutlineFlag_MissingRows	Hide missing rows: (not occurring in the other worksheet).
OutlineFlag_None	No rows/columns will be hidden.

8.1.1.8.9 ReportType

Constants used by the **Report** property in Settings.

Name	Description
ReportType_Hyperlinks	A workbook is created, range addresses are hyperlinks to the original document.
ReportType_None	No report is created.
ReportType_Standard	A workbook is created, range addresses are plain text.

8.1.1.8.10 ResultType

Constants to be used for the **ResultType** property in Settings.

Name	Description
ResultType_DifferentCom	Number or detail information of different comments.

ment	
ResultType_DifferentContent	Number of different (entered) cell values/formulas.
ResultType_DifferentFormat	Number of different formats.
ResultType_DifferentName	Number of different Excel names.
ResultType_DifferentValue	Number of different calculated cell values (results).
ResultType_DuplicateKey	Number of duplicated primary keys.
ResultType_DuplicateRec	Number of duplicated or redundant records.
ResultType_MissingCol	Number of different columns.
ResultType_MissingRow	Number of different rows.
ResultType_MissingWks	Number of missing worksheets.

8.1.1.8.11 sideID

Constants to identity the member of a set or pair.

Name	Description
sideID_src	Selects the 1st member (Source) of the set. NOTE: value is **0**).
sideID_tgt	Selects the 2nd member (Target) of the set. NOTE: value is **1**).

8.1.1.8.12 WSOptionFlag

Constants used by the **WSOptionFlag** property in Pair.

Name	Description
WSOptionFlag_Cols1on1	Columns will be linked 1 on 1 without finding inserted/deleted columns.
WSOptionFlag_None	Linking is disabled.
WSOptionFlag_Rows1on1	Rows will be linked 1 on 1 without finding inserted/deleted rows.

8.1.2 VBA Helper Functions

```
Option Explicit


'***************************************************************
'*                                                             *
'*                    SYNKRONIZER 11                           *
'*              HELPER PROCEDURES/FUNCTIONS                    *
'*                                                             *
'*  - requires a DEVELOPER license of Synkronizer 11           *
'*  - requires a reference to 'Synkronizer 11 Object Library   *
'*                                                             *
'***************************************************************


Public snk As Synkronizer.Application


'-------------------------------------------------------------
'this wrapper procedure initializes the Synkronizer application object

Public Sub InitSnk(Optional bWithUI As Boolean = False)

  If snk Is Nothing Then
    Dim cai As COMAddIn
    Set cai = Application.COMAddIns("Synkronizer.Addin")
    If Not cai.Connect Then
      'see documentation if following throws permission errors

      'Note:
      'During setup the Addin is installed for All Users by writing
      'the instructions to the registry under
      'HKLM\Software\Microsoft\Office\Excel\Addins

      'Your code should always test that the ComAddin.
      'Connect property returns TRUE.

      'If it returns FALSE, then you can activate the addin
      'from Comaddin Dialog: If you want your code to change
      'the ComAddin.Connect property, then it must be loaded
      'from HKCU (Current user). To make the necessary registry
      'changes we've provided the AddinLoad.bat batch file
      'in the installation folder.

      cai.Connect = True
    End If
    Set snk = cai.Object
  End If

  'Ensure any existing project is silently closed
  If Not bWithUI And snk.DisplayUI Then
    'hiding the UI will dispose the project
    snk.DisplayUI = False
  ElseIf bWithUI And Not snk.DisplayUI Then
    snk.DisplayUI = True
  End If

End Sub


'-------------------------------------------------------------
'this procedure closes a project
```

```
Public Sub CloseProject()
  If Not snk.ActiveProject Is Nothing Then
    If Not snk.ActiveProject.IsDisposed Then
      snk.ActiveProject.Close CloseFiles:=False, DisplayUndo:=False
    End If
  End If
End Sub



'-----------------------------------------------------
'this procedure checks if the defined
'folders & files are valid

Public Sub Check_Folders_File()

  'check folders
  If ROOT <> "" Then Debug.Assert Len(Dir(ROOT, vbDirectory)) > 0
  If FOLDERSRC <> "" Then Debug.Assert Len(Dir(FOLDERSRC, vbDirectory)) > 0
  If FOLDERTGT <> "" Then Debug.Assert Len(Dir(FOLDERTGT, vbDirectory)) > 0
  If FOLDERREP <> "" Then Debug.Assert Len(Dir(FOLDERREP, vbDirectory)) > 0
  If FOLDERPRJ <> "" Then Debug.Assert Len(Dir(FOLDERPRJ, vbDirectory)) > 0
  If FOLDERLOG <> "" Then Debug.Assert Len(Dir(FOLDERLOG, vbDirectory)) > 0

  'check files
  If FILESRC <> "" Then Debug.Assert Len(Dir(FILESRC)) > 0
  If FILETGT <> "" Then Debug.Assert Len(Dir(FILETGT)) > 0
  If PROTSRC <> "" Then Debug.Assert Len(Dir(PROTSRC)) > 0
  If PROTTGT <> "" Then Debug.Assert Len(Dir(PROTTGT)) > 0

End Sub


'----------------------------------------------------------------
'this function returns the not matched files

Public Function Get_NotMatchedWorksheets(sSrcFolder As String,
sTgtFolder As String) As Variant
  ReDim aFiles(0) As String
  ReDim aNotMatched(0) As String
  Dim sFile As String
  Dim i, j As Integer

  For i = 1 To 2

    sFile = Dir(Choose(3 - i, sSrcFolder, sTgtFolder) & "*.xls*")
    Do While Len(sFile) > 0
      ReDim Preserve aFiles(UBound(aFiles) + 1)
      aFiles(UBound(aFiles)) = sFile
      sFile = Dir
    Loop
    For j = 1 To UBound(aFiles)
      If Len(Dir(Choose(i, sSrcFolder, sTgtFolder) & aFiles(j))) = 0
Then
        ReDim Preserve aNotMatched(UBound(aNotMatched) + 1)
        aNotMatched(UBound(aNotMatched)) = Choose(i, sSrcFolder,
sTgtFolder) & aFiles(j)
      End If
    Next j

    ReDim aFiles(0)
  Next i

  Get_NotMatchedWorksheets = aNotMatched

End Function


'----------------------------------------------------------------
'this function returns the differences of a project
'if on pair is compared, all detailed differences are returned
'if multiple pairs are compared, the total differences per pair are
returned

Public Function GetDifferences(oProj As Project) As String
  Dim p As Pair
  Dim sMsg As String
  Dim i As Integer

  If oProj.Pairs.Count = 1 Then
    'one pair » return detailed differences
    sMsg = oProj.Results.SumText

  Else
    'multiple pairs » return total differences per pair
    i = 1
    For Each p In oProj.Pairs
      sMsg = sMsg & p.SheetName(sideID_src) & vbTab & _
             p.Results.Sum & vbNewLine
      i = i + 1
      If i > 20 Then Exit For
    Next p

    'just display the first 20 pairs...
    If i > 20 Then
      sMsg = sMsg & "..." & vbNewLine
    End If
  End If

  GetDifferences = sMsg
End Function
```

```
'---------------------------------------------------------------
'this function compares the files of two folders
'
'Parameter description
'Paramater description:
'sFolderSrc:  Folder with source files to be compared
'sFolderTgt:  Folder with target files to be compared
'bHighlight:  Select True if differences should be highlighted
'sFolderLog:  If difference reports are needed, enter folder. Optional.
'sFolderLog:  If a log file is needed, enter folder. Optional

Public Function SynkFolders(sFolderSrc As String, _
                            sFolderTgt As String, _
                            bHighlight As Boolean, _
                            Optional sFolderRep As String, _
                            Optional sFolderLog As String) As String

  Dim oProj As Synkronizer.Project
  Dim sFile As String
  Dim aFiles() As String
  Dim i As Integer
  Dim j As Integer
  Dim sFileSrc As String
  Dim sFileTgt As String
  Dim sFileRep As String
  Dim sFileLog As String
  Dim vNotMatchedFiles As Variant
  Dim n(0 To 1) As Long
  Dim t0 As Date

  'check if folders are valid
  Debug.Assert Len(Dir(sFolderSrc, vbDirectory))
  Debug.Assert Len(Dir(sFolderTgt, vbDirectory))
  If sFolderRep <> "" Then Debug.Assert Len(Dir(sFolderRep,
vbDirectory))
  If sFolderLog <> "" Then Debug.Assert Len(Dir(sFolderLog,
vbDirectory))

  t0 = Timer

  On Error GoTo theExit

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create log file
  If sFolderLog <> "" Then
    sFileLog = sFolderLog & "\synkronizer_log_" & Format(Now, "yyyy-mm-
dd_HH-MM-SS") & ".txt"
    Reset
    Open sFileLog For Output As #1
    Print #1, "Synkronizer Logfile"
    Print #1, "-------------------"
    Print #1, ""
    Print #1, "Date: " & Format(Date, "yyyy-mm-dd")
    Print #1, "Time: " & Format(Time, "hh:nn:ss")
    Print #1, ""
    Print #1, ""
  End If


  'read "source" files
  i = 0
  sFile = Dir(sFolderSrc & "*.xls*")
  Do While Len(sFile) > 0
    ReDim Preserve aFiles(i)
    aFiles(i) = sFile
    i = i + 1
    sFile = Dir
  Loop


  'log not matched worksheets
  vNotMatchedFiles = Get_NotMatchedWorksheets(sFolderSrc, sFolderTgt)
  If UBound(vNotMatchedFiles) > 0 Then
    Print #1, "Not matched files"
    For i = 1 To UBound(vNotMatchedFiles)
      Print #1, vNotMatchedFiles(i)
    Next i
    Print #1, ""
    Print #1, ""
  End If


  'loop all "source" files
  For i = 0 To UBound(aFiles)
    sFileSrc = sFolderSrc & aFiles(i)
    sFileTgt = sFolderTgt & aFiles(i)
    sFileRep = sFolderRep & "Difference Report " & aFiles(i)
    sFileRep = Left(sFileRep, InStrRev(sFileRep, ".") - 1) & ".xlsx"

    'check if "target" is there
    If Len(Dir(sFileTgt)) > 0 Then

      'create new project
      Set oProj = snk.NewProject

      With oProj
        'load files
        .Files.Load sFileSrc, sFileTgt
        'match all worksheets with same name
        With .Pairs
          .MatchType = MatchType_AllByName
          .MatchInclude = MatchIncludeFlag_HiddenSheets +
MatchIncludeFlag_ProtectedSheets
          .AddMatched
        End With

        'highlight & create report
        With .Settings
          If bHighlight Then .Highlight = HighlightType_Standard
          If sFolderRep <> "" Then .Report = ReportType_Standard
        End With

        'compare!
        .Execute

        'log differences
        If sFolderLog <> "" Then
          'Print #1, aFiles(i) & vbTab & .Results.Sum
          Call Logfile_PrintDiffs(oProj)
        End If

        If .Results.Sum Then
          'if differences found, create report
          n(1) = n(1) + 1

          If sFolderRep <> "" Then
            If Len(Dir(sFileRep)) > 0 Then Kill sFileRep
            With .ReportWorkbook
              .SaveAs file name:=sFileRep
            End With
          End If
        Else
          'no differences noted; close report without saving
          n(0) = n(0) + 1

        End If

        'save files if differences are highlighted
        If bHighlight Then
          If .Files.Workbook(sideID_src).FullName <> sFileSrc Then
            .Files.Workbook(sideID_src).SaveCopyAs sFileSrc
          Else
            .Files.Workbook(sideID_src).Save
          End If
          If .Files.Workbook(sideID_tgt).FullName <> sFileTgt Then
            .Files.Workbook(sideID_tgt).SaveCopyAs sFileTgt
          Else
            .Files.Workbook(sideID_tgt).Save
          End If
        End If
        .Close CloseFiles:=True, DisplayUndo:=False
        DoEvents
      End With

      Set oProj = Nothing
      DoEvents
    End If
  Next i

  'create end message in log file
  If sFolderLog <> "" Then
    Print #1, ""
    Print #1, "Comparison time: " & Format(Timer - t0, " 00.00\s\")
    Reset
  End If

  'display end message
  SynkFolders = "finished" & vbLf & _
                n(0) & " workbooks without differences" & vbLf & _
                n(1) & " workbooks with differences, see reports"

theExit:
  Reset
  Set oProj = Nothing
  Set snk = Nothing
  Exit Function

theError:
  Dim sErr As String
  sErr = Err.Number & ": " & Err.Description
  On Error Resume Next
  If Not oProj Is Nothing Then
    oProj.Close True, False
  End If

  SynkFolders = sErr
  Resume theExit
End Function


'----------------------------------------------------------------
'this function compares one source file against all files of a target
folder
'
'Paramater description:
'sFileSrc:    Source file
'sFolderTgt:  Folder with target files to be compared
'bHighlight:  Select True if differences should be highlighted
'sFolderLog:  If difference reports are needed, enter folder. Optional.
'sFolderLog:  If a log file is needed, enter folder. Optional

Public Function SynkSrcFolder(sFileSrc As String, _
                              sFolderTgt As String, _
                              bHighlight As Boolean, _
                              Optional sFolderRep As String, _
                              Optional sFolderLog As String) As String

  Dim oProj As Synkronizer.Project
  Dim aFiles() As String
  Dim i As Integer
  Dim sFile As String
  Dim sFileTgt As String
  Dim sFileRep As String
  Dim sFileLog As String
  Dim n(0 To 1) As Long
  Dim t0 As Date

  'check if files/folders are valid
  Debug.Assert Len(Dir(sFileSrc, vbDirectory))
  Debug.Assert Len(Dir(sFolderTgt, vbDirectory))
  If sFolderRep <> "" Then Debug.Assert Len(Dir(sFolderRep,
vbDirectory))
  If sFolderLog <> "" Then Debug.Assert Len(Dir(sFolderLog,
vbDirectory))

  t0 = Timer

  On Error GoTo theError

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create log file
  If sFolderLog <> "" Then
    sFileLog = sFolderLog & "\synkronizer_log_" & Format(Now, "yyyy-mm-
dd_HH-MM-SS") & ".txt"
    Reset
    Open sFileLog For Output As #1
    Print #1, "Synkronizer Logfile"
    Print #1, "-------------------"
    Print #1, ""
    Print #1, "Date: " & Format(Date, "yyyy-mm-dd")
    Print #1, "Time: " & Format(Time, "hh:nn:ss")
    Print #1, ""
    Print #1, ""
  End If


  'read "target" files
  i = 0
  sFile = Dir(sFolderTgt & "*.xls*")
  Do While Len(sFile) > 0
```

```
    ReDim Preserve aFiles(i)
    aFiles(i) = sFile
    i = i + 1
    sFile = Dir
  Loop

  'loop all files
  For i = 0 To UBound(aFiles)
    sFileTgt = sFolderTgt & aFiles(i)
    sFileRep = sFolderRep & "Difference Report " & aFiles(i)
    sFileRep = Left(sFileRep, InStrRev(sFileRep, ".") - 1) & ".xlsx"

    'create new project
    Set oProj = snk.NewProject

    With oProj
      'load files
      .Files.Load sFileSrc, sFileTgt

      'match all worksheets with same name
      With .Pairs
        .MatchType = MatchType_AllByName
        .MatchInclude = MatchIncludeFlag_HiddenSheets +
MatchIncludeFlag_ProtectedSheets
        .AddMatched
      End With

      'highlight & create report
      With .Settings
        If bHighlight Then .Highlight = HighlightType_Standard
        If sFolderRep <> "" Then .Report = ReportType_Standard
      End With

      'compare!
      .Execute

      'log differences
      If sFolderLog <> "" Then
        'Print #1, aFiles(i) & vbTab & .Results.Sum
        Call Logfile_PrintDiffs(oProj)
      End If

      If .Results.Sum Then

        'if differences found, create report
        n(1) = n(1) + 1

        If sFolderRep <> "" Then
          If Len(Dir(sFileRep)) > 0 Then Kill sFileRep
          If Not .ReportWorkbook Is Nothing Then
            With .ReportWorkbook
              .SaveAs file name:=sFileRep
            End With
          End If
        End If
      Else
        'no differences noted; close report without saving
        n(0) = n(0) + 1
      End If

      'save target file if differences are highlighted
      If bHighlight Then

        If .Files.Workbook(sideID_tgt).FullName <> sFileTgt Then
          .Files.Workbook(sideID_tgt).SaveCopyAs sFileTgt
        Else
          .Files.Workbook(sideID_tgt).Save
        End If

        '.Files.Workbook(sideID_tgt).Save
      End If

      .Close CloseFiles:=True, DisplayUndo:=False
      DoEvents
    End With

    Set oProj = Nothing
  Next i

  'return value
  SynkSrcFolder = "finished" & vbLf & _
                  n(0) & " workbooks without differences" & vbLf & _
                  n(1) & " workbooks with differences, see reports"



  'write end message in log file
  If sFolderLog <> "" Then
    Print #1, ""
    Print #1, "Comparison time: " & Format(Timer - t0, " 00.00\s\")
    Reset
  End If


theExit:
  Reset
  Set oProj = Nothing
  Set snk = Nothing
  Exit Function

theError:
  Dim sErr As String
  sErr = Err.Number & ": " & Err.Description
  On Error Resume Next
  If Not oProj Is Nothing Then
    oProj.Close True, False
  End If

  SynkSrcFolder = sErr
  Resume theExit
End Function


'-----------------------------------------------------------------
'this procedure writes all project differences in a new workbook

Public Sub DumpDetails_Project(oProj As Synkronizer.Project)
  Dim wkb As Workbook
  Dim wks As Worksheet
  Dim val As Variant
  Dim rng As Range
  Dim p As Pair
  Dim iWksCount As Integer

  Debug.Assert Not oProj Is Nothing
  Debug.Assert Not oProj.IsDisposed

  'create workbook
  iWksCount = Application.SheetsInNewWorkbook
  Application.SheetsInNewWorkbook = 1
  Set wkb = Workbooks.Add
  Application.SheetsInNewWorkbook = iWksCount

  Set wks = wkb.Worksheets(1)

  'name worksheet
  wks.Name = "Project Differences"

  val = oProj.Results.ArrayDetails
  If wks.UsedRange.Cells.CountLarge = 1 Then
    Set rng = wks.Cells(1)
  Else
    With wks.UsedRange
    Set rng = wks.Cells(.Row + .Rows.Count, .Column)
    End With
  End If

  If IsEmpty(val) Then
    rng.Offset(0, 0).Value = "no diffs"
  Else
    Set rng = rng.Offset(0, 0).Resize(UBound(val, 1) + 1, UBound(val, 2)
+ 1)
    rng.Clear
    rng.Resize(, 11).NumberFormat = "@"
    rng.Resize(, 4).HorizontalAlignment = xlLeft
    rng.VerticalAlignment = xlTop
    rng.Value2 = val

  End If

  'format range
  With rng
    .Rows(1).Font.Bold = True

    .Columns("I:K").HorizontalAlignment = xlRight

    .Columns("A").ColumnWidth = 20
    .Columns("B:K").ColumnWidth = 8
    .Columns("D:E").ColumnWidth = 32
    .Columns("F:H").ColumnWidth = 18

    .Columns("F:G").EntireColumn.Hidden = True
    For Each p In oProj.Pairs
      If p.DBKeys <> "" Then
        .Columns("F:G").EntireColumn.Hidden = False
        Exit For
      End If
    Next p
  End With

  DoEvents
  Application.ScreenUpdating = True

End Sub

'---------------------------------------------------------------
'this procedure creates a new workbook and
'writes all pair differences in a separate worksheet

Public Sub DumpDetails_Pairs(oProj As Synkronizer.Project)

  Dim wkb As Workbook
  Dim wks As Worksheet
  Dim p As Pair
  Dim iPair As Integer
  Dim val As Variant
  Dim rng As Range
  Dim iWksCount As Integer

  'check if project is active
  Debug.Assert Not oProj Is Nothing
  Debug.Assert Not oProj.IsDisposed

  'create workbook
  iWksCount = Application.SheetsInNewWorkbook
  Application.SheetsInNewWorkbook = oProj.Pairs.Count
  Set wkb = Workbooks.Add
  Application.SheetsInNewWorkbook = iWksCount
  'loop through pairs
  iPair = 1
  For Each p In oProj.Pairs

    'name worksheet
    Set wks = wkb.Worksheets(iPair)
    wks.Name = p.SheetName(0)

    'get results
    val = p.Results.ArrayDetails
    Set rng = wks.Cells(1)

    'write title
    With rng
      .Value = p.SheetName(0)
      .Font.Size = 12
      .Font.Bold = True
    End With

    'write down differences
    If IsEmpty(val) Then
      'no differences found
      rng.Offset(2, 0).Value = "no diffs"
    Else
      'differences found
      Set rng = rng.Offset(2, 0).Resize(UBound(val, 1) + 1, UBound(val,
2) + 1)
      rng.Clear
      rng.Resize(, 6).NumberFormat = "@"
      rng.Resize(, 2).HorizontalAlignment = xlLeft
      rng.VerticalAlignment = xlTop
      rng.Value2 = val
    End If
    val = Empty

    'format range
    With rng
      .Rows(1).Font.Bold = True

      .Columns("I:K").HorizontalAlignment = xlRight

      .Columns("A").ColumnWidth = 20
      .Columns("B:K").ColumnWidth = 8
      .Columns("D:E").ColumnWidth = 32
      .Columns("F:H").ColumnWidth = 18

      If p.DBKeys = "" Then
        .Columns("F:H").EntireColumn.Hidden = True
      End If
    End With

    iPair = iPair + 1
  Next p

  DoEvents
  Application.ScreenUpdating = True
End Sub


'---------------------------------------------------------------
'this procedure creates a log file with all differences

Public Sub Logfile_PrintDiffs(oProj As Synkronizer.Project)
  Dim p As Pair
  Dim sText As String
  Dim sLine As String


  'source file
  sText = "Source File"
  sLine = sText & String(34 - Len(sText), " ") & oProj.Files(sideID_src)
  Print #1, sLine

  'target file
  sText = "Target File"
  sLine = sText & String(34 - Len(sText), " ") & oProj.Files(sideID_tgt)
  Print #1, sLine

  '--------------------------------------------------------------
  'heading row differences

  sText = "Worksheet names"
  sLine = sText & String(32 - Len(sText), " ")
  sLine = sLine & "  Total"
  sLine = sLine & _
          "  Columns" & _
          "  DupRecs" & _
          "  DupKeys" & _
          "     Rows" & _
          " Contents" & _
          "   Values"
  If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
    sLine = sLine & "  Formats"
  End If
  If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
    sLine = sLine & " Comments"
  End If
  If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
    sLine = sLine & "    Names"
  End If

  Print #1, sLine
```

```
'--------------------------------------------------------------
'pair differences

For Each p In oProj.Pairs
  With p.Results
    sText = p.SheetName(sideID_src)
    sLine = sText & String(32 - Len(sText), " ")

    sText = CStr(.Sum)
    sLine = sLine & String(7 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingCol)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateRec)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateKey)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingRow)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentContent)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentValue)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
      sText = .SumTextByType(ResultType_DifferentFormat)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
      sText = .SumTextByType(ResultType_DifferentComment)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
      sText = .SumTextByType(ResultType_DifferentName)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    Print #1, sLine
  End With
Next p


'--------------------------------------------------------------
'total differences

If oProj.Pairs.Count > 1 Then
```

```
    With oProj.Results
      sText = "Total"
      sLine = sText & String(32 - Len(sText), " ")

      sText = CStr(.Sum)
      sLine = sLine & String(7 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_MissingCol)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_DuplicateRec)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_DuplicateKey)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_MissingRow)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_DifferentContent)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      sText = .SumTextByType(ResultType_DifferentValue)
      sLine = sLine & String(9 - Len(sText), " ") & sText

      If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
        sText = .SumTextByType(ResultType_DifferentFormat)
        sLine = sLine & String(9 - Len(sText), " ") & sText
      End If

      If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
        sText = .SumTextByType(ResultType_DifferentComment)
        sLine = sLine & String(9 - Len(sText), " ") & sText
      End If

      If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
        sText = .SumTextByType(ResultType_DifferentName)
        sLine = sLine & String(9 - Len(sText), " ") & sText
      End If

      Print #1, sLine
    End With
  End If

  Print #1, ""
  Print #1, ""

End Sub
```

8.1.3 VBA Examples

```
Option Explicit


'*******************************************************************
'*                                                                 *
'*                    SYNKRONIZER 11                               *
'*                     VBA EXAMPLES                                *
'*                                                                 *
'*  To test the examples you need                                  *
'*  - to have a DEVELOPER license of Synkronizer 11                *
'*  - to create a reference to 'Synkronizer 11 Object Library'     *
'*  - include the module 'snk_helper_functions' which contains     *
'*    helper procedures and functions                              *
'*                                                                 *
'*******************************************************************


'define folders
Public Const ROOT As String = "D:\Documents\"
Public Const FOLDERSRC As String = ROOT & "Source\"
Public Const FOLDERTGT As String = ROOT & "Target\"
Public Const FOLDERREP As String = ROOT & "Reports\"
Public Const FOLDERPRJ As String = ROOT & "Projects\"
Public Const FOLDERLOG As String = ROOT & "Log\"

'define files
Public Const FILESRC As String = FOLDERSRC & "Source 1.xlsx"
Public Const FILETGT As String = FOLDERTGT & "Target 2.xlsx"
Public Const FILEREP As String = FOLDERREP & "Synkronizer_Difference_Report.xlsx"
Public Const FILEPRJ As String = FOLDERPRJ & "SynkProject.xml"
Public Const PROTSRC As String = FOLDERSRC & "Protected 1.xlsx"
Public Const PROTTGT As String = FOLDERTGT & "Protected 2.xlsx"


'-----------------------------------------------------------
'Example 1
'- compare all sheets with the same name
'- create a difference report
'- highlight differences
'- show all differences except identical rows
'- create a project with all settings

Public Sub Example1()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File
```

```
'get access to the Synkronizer application object
InitSnk

'create project
Set oProj = snk.NewProject
With oProj

  'load files
  .Files.Load FILESRC, FILETGT

  'define worksheets
  With .Pairs
    .MatchType = MatchType_AllByName
    .AddMatched
  End With

  With .Settings

    'create a report
    .Report = ReportType_Standard

    'highlight differences
    .Highlight = HighlightType_WithReset

    'show only rows/columns with differences
    .Outline = OutlineFlag_Enabled + _
               OutlineFlag_DifferentRows + OutlineFlag_MissingRows + OutlineFlag_
               OutlineFlag_MissingCols

    'also compare comments and names
    .Contents = ContentFlag_Comments + ContentFlag_Names

    'also compare also font formats
    .Formats = FormatFlag_Enabled + FormatFlag_Font

  End With

  'start Synkronizer
  .Execute

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'save project
  If snk.DisplayUI = False Then
    .Save (FILEPRJ)
  End If

  'close project
  CloseProject

End With
```

```
  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 2
'- compare a project

Public Sub Example2()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'load Synkronizer project
  Set oProj = snk.OpenProject(FILEPRJ)

  'check if source & target files are valid
  Debug.Assert oProj.Files.IsValid

  'start Synkronizer
  oProj.Execute

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'close project
  CloseProject

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 3
```

```
'- compare protected sheets
'- highlight differences.

Public Sub Example3()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load PROTSRC, PROTTGT

    With .Pairs

      'add worksheets
      .AddPair "Customer List", "Customer List"
      .AddPair "Customer Profile", "Customer Profile"

      'enter passwords (separated by semicolons)
      .PasswordList = "abc;def"
   End With

    'highlight differences
    With .Settings
      .Highlight = HighlightType_Standard
    End With

    'start Synkronizer
    .Execute

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub
```

```
Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'-----------------------------------------------------------
'Example 4
'- compare first sheets
'- compare formats, comments, names & use filters
'- create a report

Public Sub Example4()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load FILESRC, FILETGT

    'define first worksheet of each file
    With .Pairs
      .MatchType = MatchType_FirstByName
      .AddMatched
    End With

    With .Settings

      'also compare comments and names
      .Contents = ContentFlag_Comments + ContentFlag_Names

      'compare also font formats
      .Formats = FormatFlag_Enabled + FormatFlag_Font

      'use some filters
      .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpac
      .FilterTolerance = 0.01
      .FilterEquivalents = "yes,ja;no,nein"

      'create a report
      .Report = ReportType_Standard
    End With

    'compare files
    .Execute

    If Not .ReportWorkbook Is Nothing Then

      'delete report if it already there
      If Len(Dir(FILEREP)) > 0 Then Kill FILEREP

      'save report
      With .ReportWorkbook
        .SaveAs file name:=FILEREP
        .Close SaveChanges:=False
      End With
    End If

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 5
'- database comparison
'- highlight differences

Public Sub Example5()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load FILESRC, FILETGT

    'define worksheets & database options
    .Pairs.AddPair Sheet0:="Customer List", _
                   Sheet1:="Customer List", _
                   DBRow:=1, _
                   DBKeys:="1;2", _
                   DBOptions:=DBOptionFlag_Group

    'highlight differences
    With .Settings
      .Highlight = HighlightType_WithReset
    End With

    'start Synkronizer
    .Execute

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 6
'- link rows 1 on 1
'- highlight differences

Public Sub Example6()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File
  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load FILESRC, FILETGT

    'define worksheets & link options
    .Pairs.AddPair Sheet0:="Customer List", _
                   Sheet1:="Customer List", _
                   WSOptions:=WSOptionFlag_Rows1on1

    'highlight differences
    With .Settings
      .Highlight = HighlightType_WithReset
    End With

    'start Synkronizer
    .Execute

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'-----------------------------------------------------------
'Example 7
'- compare a project
'- create workbook which contains all differences

Public Sub Example7()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example
  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'load Synkronizer project
  Set oProj = snk.OpenProject(FILEPRJ)

  'check if source & target files are valid
  Debug.Assert oProj.Files.IsValid

  'start Synkronizer
  oProj.Execute

  'create a workbook with all detail differences
  DumpDetails_Project oProj

  'create a workbook with all detail differences
  'all pair differences will be written in a separate worksheet
  DumpDetails_Pairs oProj

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'close project
  CloseProject

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'-----------------------------------------------------------
'Example 8
'- compare all Excel files with the same names of two folders
'- a difference report will be created of each file set
'- a log file will be created

Public Sub Example8()
  Dim sMsg As String

  'Paramater description:
  'sFolderSrc:  Folder with source files to be compared
  'sFolderTgt:  Folder with target files to be compared
  'bHighlight:  Select True if differences should be highlighted
  'sFolderLog:  If difference reports are needed, enter folder. Optional.
  'sFolderLog:  If a log file is needed, enter folder. Optional

  sMsg = SynkFolders(sFolderSrc:=FOLDERSRC, _
                     sFolderTgt:=FOLDERTGT, _
                     bHighlight:=True, _
                     sFolderRep:=FOLDERREP, _
                     sFolderLog:=FOLDERLOG)

  MsgBox sMsg

End Sub


'------------------------------------------------------------
'Example 9
'- compare one "source" file against a series of target files
'- a difference report will be created for each file set
'- a log file will be created

Public Sub Example9()
  Dim sMsg As String

  'Paramater description:
  'sFileSrc:    Source file
  'sFolderTgt:  Folder with target files to be compared
  'bHighlight:  Select True if differences should be highlighted
  'sFolderLog:  If difference reports are needed, enter folder. Optional.
  'sFolderLog:  If a log file is needed, enter folder. Optional

  sMsg = SynkSrcFolder(sFileSrc:=FILESRC, _
                       sFolderTgt:=FOLDERTGT, _
                       bHighlight:=True, _
                       sFolderRep:=FOLDERREP, _
                       sFolderLog:=FOLDERLOG)

  MsgBox sMsg

End Sub
```

8.2 CommandLine Utility

With the Developer Edition of Synkronizer 11, it is also possible to compare Excel files via the **command line commands**. **The Excel files can be compared fully automated with a single mouse click - without opening Excel!**

How to use the CommandLine Utility?

synk.exe is a handy program which allows you to compare Excel files without opening Excel. You can enter all variables like file names, folders and comparison options in a single command. The program then generates different reports and log files with all differences. While executing, the program will start a hidden instance of Excel. After the comparison process the Excel instance will be closed again.

Which tasks does the CommandLine Utility support?

The following tasks are supported:

- Compare two files
- Compare a source file against a bundle of source files
- Compare all Excel files (with equal file names) in two folders
- Create a difference report
- Create a log file which contains all differences
- Format and filter options are supported

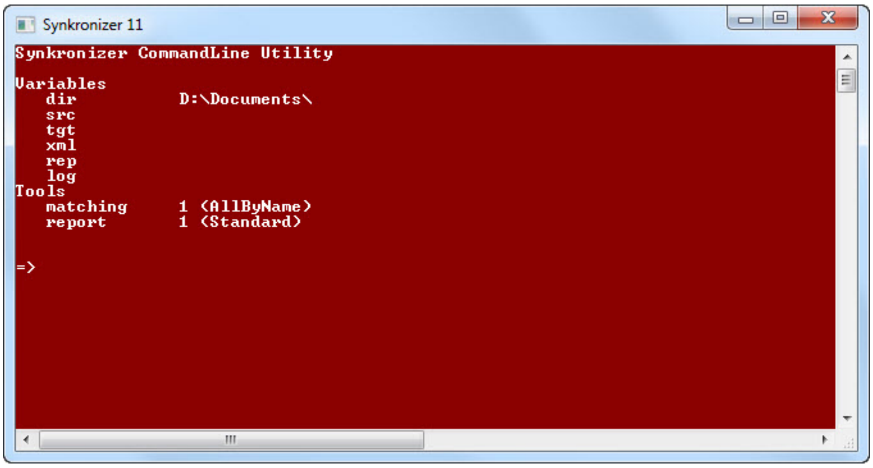
Which tasks are not supported by the CommandLine Utility?

The following tasks are not supported:

- Source and target files cannot be saved
- Differences cannot be highlighted
- Differences cannot be outlined (shown/hidden)

The CommandLine Utility of Synkronizer is invoked as follows:

1. Start the Windows Explorer
2. Select the Synkronizer folder
   `c:\Program Files\Synkronizer\Synkronizer 11\`
3. Double-click on **synk.exe**
4. The Sykronizer CommandLine Utility appears.



```
Synkronizer CommandLine Utility

Variables
   dir          D:\Documents\
   src
   tgt
   xml
   rep
   log
Tools
   matching     1 (AllByName)
   report       1 (Standard)

=>
```

Start CommandLine Utility with a batch file or Windows shortcut

We recommend that you create a batch file or a Windows shortcut. In this way you can start the CommandLine Utility with a single mouse click.

Creating a batch file

Proceed as follows:

1. Start a text editor and create a new document.
2. Write the following code:

```
@echo off
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe"
```

3. Save the file as synk.bat in a folder of preference

Creating a Windows shortcut

You can also create a Windows shortcut. Proceed as follows:

1. Right-click an open area on the desktop, point to New, and then click Shortcut.
2. Click Browse.
3. Locate **synk.exe** which is stored in the following folder:
   `C:\Program Files\Synkronizer\Synkronizer 11\`
   click Open, and then click Next.
4. Type a name for the shortcut and click Finish button.
5. Right-click on the created shortcut.
6. Enter in the field "Start in" the folder, in which the Synkronizer files should be stored.
7. Click OK and close the shortcut

8.2.1 Reference

This page contains the syntax of the command line utility and a description of all variables, commands and settings.

Syntax

```
synk[.exe]
  [/dir={Parent folder}]
  [/src={Source folder/file}]
  [/tgt={Target folder/file}]
  [/xml={Synkronizer project file}]
  [/rep={Report folder}]
  [/log={Log file}]
  [/m={Matching options}]
  [/h={Highlighting options}]
  [/r={Report options}]
  [/c]
  [/x]
```

Variables

/dir={Parent folder}
Optional. Name/location of the parent folder. The difference reports and log files will be stored within this folder if not defined separately.

/src={Source folder/file}
Required. Folder of the **source files** or the name of the **source file**. If you want to compare all files of a folder you need to enter a folder otherwise enter a file name. You can enter an absolute path/file or a path/file name relative to the parent directory (/dir).

/tgt={Target folder/file}
Required. Folder of the **target files** or the name of the **target file**. If you want to compare all files of a folder you need to enter a folder otherwise enter a file name. You can enter an absolute path/file or a path/file name relative to the parent directory (/dir).

/xml={Synkronizer Project file}
Optional. Name of the Synkronizer project file. All other settings like Contents, Formats or Filters are taken from this project file. If no project file is defined, the Synkronizer default settings are used.

/rep={Folder for difference reports}
Optional. Name of the folder in which the difference reports are saved. If this parameter is empty, the difference reports are written into the parent folder (/dir).

/log={Name of Synkronizer log file}
Optional. Name of the log file. If this parameter is defined, a log file will be created.

Note
If the folders or file names contain **spaces** and you are working with **batch files** and/or **Windows shortcuts** you need to use **apostrophes** (double quotes) for addressing the folders/ file names.

Tools

matching (/m=)
Optional. Matching options. You can define how the worksheets shall be matched. The following options are allowed:
1=Compare all worksheets by name (default setting)
2=Compare all worksheets by index
-1=Compare first worksheets by name
-2=Compare first worksheets by index

highlight (/h=)
Optional. Highlighting options. The following options are allowed:
1=Differences are highlighted
0=No differences are highlighted (default setting)

report (/r=)
Optional. Difference report options. The following options are allowed:
1=A standard difference report is created (default setting)
2=A difference with hyperlinks will be created.
0=No difference report will be created.

Commands

/c
Starts the comparison process. You can enter ctrl-c to terminate the comparison process.

/cs
Starts the comparison process and saves the files at the end under a new name. The original remain files unchanged. You can enter ctrl-c to terminate the comparison process.

/rs
Run & show. Runs the first batch (or file pair) and opens Synkronizer and shows the results screen with the Excel files.

/x
The Synkronizer CommandLine Utility will be closed.

/f
Files to be compared. You can see which files will be compared and which not.

/v
The settings of the current project are displayed. You can see the folders, file names and comparison settings.

/cls
Clears the screen.

/dir
Shows content of directory.

/xml
Shows xml content of the project template.

/l
Loads Excel instance.

/u
Unloads Excel instance.

/xlShow
Shows the owned Excel instance (if available).

/xlHide
Hides the owned Excel instance (if available).

/?
Synkronizer CommandLine Utility help. All commands and settings are displayed.

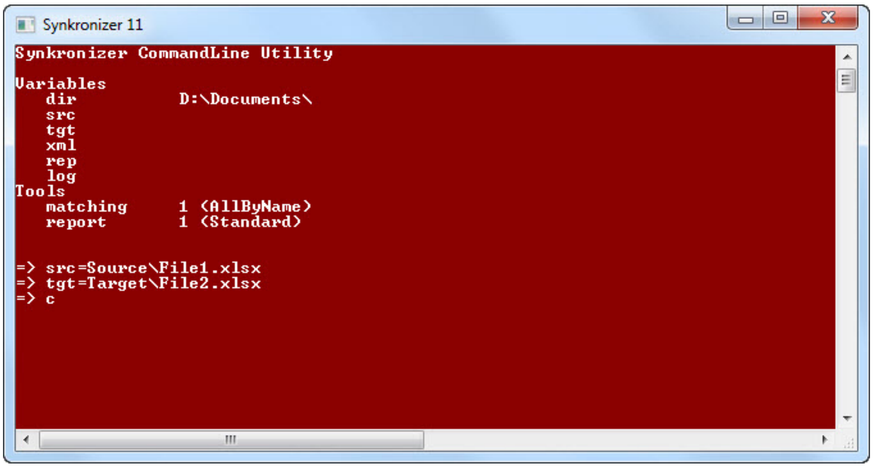
/!
Synkronizer CommandLine Utility help. Further commands and settings are shown displayed.

8.2.2 Examples

This section contains some examples on how you can compare Excel files with the Synkronizer CommandLine Utility.

Example 1 - Compare two files

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target file with the command **tgt={file name}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**



```
Synkronizer CommandLine Utility

Variables
   dir          D:\Documents\
   src
   tgt
   xml
   rep
   log
Tools
   matching     1 (AllByName)
   report       1 (Standard)


=> src=Source\File1.xlsx
=> tgt=Target\File2.xlsx
=> c
```

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src=Source
```

Example 2 - Compare two files and open Excel

This example compares two files and opens then Excel and Synkronizer. The two files and Synkronizer results screen are displayed.

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target file with the command **tgt={file name}**
4. Compare the files with the command **rs**
5. Close the Synkronizer CommandLine Utility with the command **x**

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src=Source
```

Example 3 - Compare one master file with various target files

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target folder (which contains the target files) with the command **tgt={target folder}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src=Source
```

Example 4 - Compare all files in two folders

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source folder with the command **src={source folder}**
3. Enter the target folder with the command **tgt={target folder}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src="Sourc
```

Example 5 - Create no difference report but a logfile

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target file with the command **tgt={file name}**
4. Enter the name of the logfile: **log={file name}**
5. Define report settings. Enter **r=0**
6. Compare the files with the command **c**
7. Close the Synkronizer CommandLine Utility with the command **x**

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src=Source
```

Example 6 - Compare two files using filters and formats

1. Start Excel & Synkronizer
2. Define the files, filters and formats which you'd like to compare
3. Save the settings as a project
4. Close Synkronizer and Excel
5. Start the CommandLine Utility of Synkronizer using **synk.exe**
6. Enter the source file with the command **src={file name}**
7. Enter the target file with the command **tgt={file name}**
8. Enter the Synkronizer project file which contains the settings: **xml={file name}**
9. Compare the files with the command **c**
10. Close the Synkronizer CommandLine Utility with the command **x**

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" dir=D:\Documents\ src=Source
```

Index