

Text count in AnyCount and MS Office

Introduction

AnyMem is a tool for professional translators who work with texts partially or fully repeating from job to job. The program is not a "machine translator" (professional translators never use those anyway), as it translates nothing but memorizes everything translated by the translator and offers a previously created translation when an identical or similar sentence is translated. In practice it means that the translator does not translate the same text twice but rather concentrates on the creative part of work.

Supposing a translator has five slightly varying patents or agreements to translate, then after the first document is translated only the altering parts of the remaining documents would need to be translated. This way the translator saves lots of time which can be spent on rest or additional work (thus making more money). Moreover, the memory created in translation can be passed on to a colleague for further work.

Setup

AnyMem is integrated into MS Word as a macro. Therefore MS Word should be installed prior to AnyMem installation. After AnyMem is installed the following toolbar appears in MS Word:



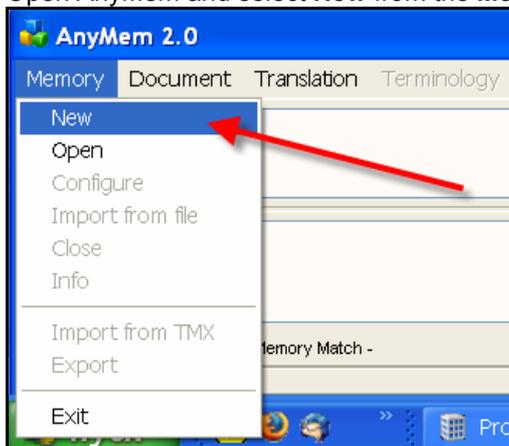
All translation process is carried out with the help of toolbar buttons. An hour would suffice to learn the basics of this program.

слово

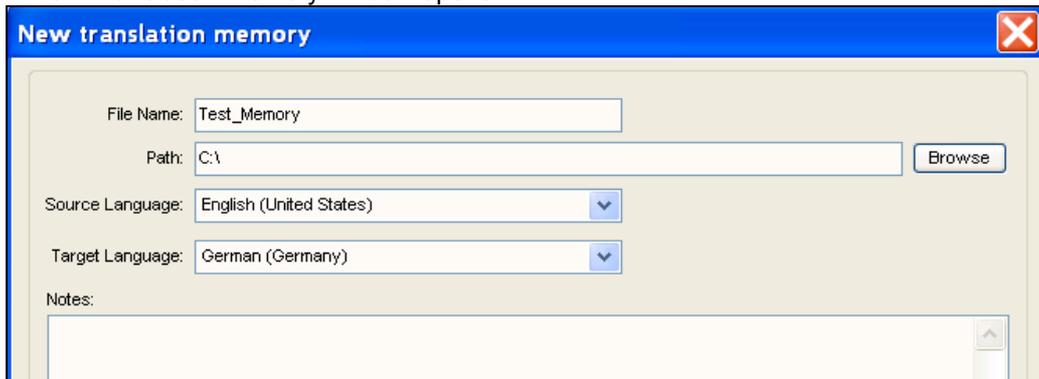
Getting started

Before you start working, you need to create a memory file. The file is created as follows:

1. Open AnyMem and select **New** from the **Memory** menu:



2. A New Translation Memory window opens:



3. Enter a desired memory file name in the **File Name** field, select location for the memory file in the **Path** field (use the **Browse** button if necessary).
4. Select the **Source language** and the **Target language**.
5. Click **OK**.

Thus the memory file is created and you can start working. Let us examine program operation with a simple example. Suppose the following text should be translated:

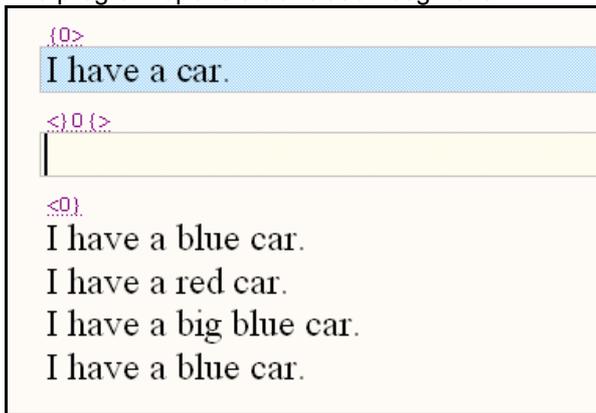
I have a car.
I have a blue car.
I have a red car.
I have a big blue car.
I have a blue car.

To start working:

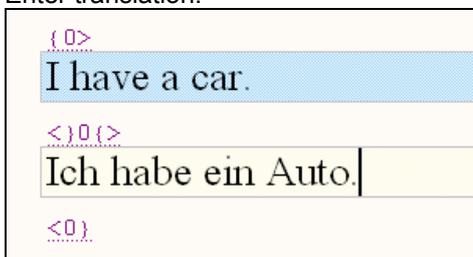
1. Place the pointer on the first of the sentences to translate (I have a car) and click **Start Translation**:



2. The program opens a translation segment:



3. Enter translation:

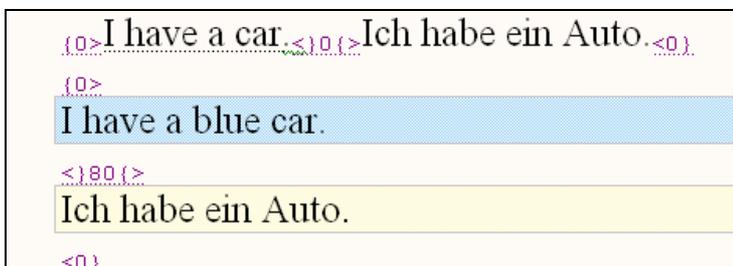


Attention! Translation should be entered strictly in the yellow translation field. Be careful not to damage segmentation markers `{0>` (beginning of a segment) `<0{>` (separation of the source and translation text) `<0}` (end of the translated segment).

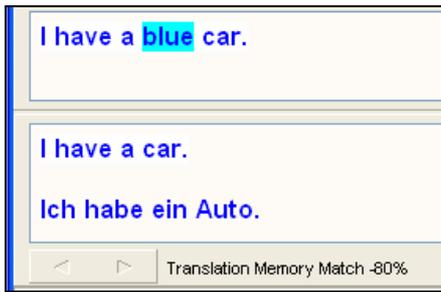
4. After translation is completed, click the **Next Segment** button:



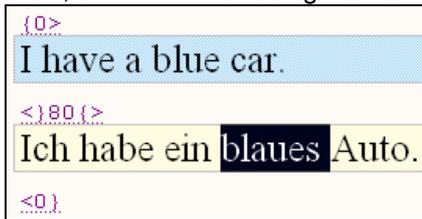
5. The program opens the next translation segment:



As can be seen below, the program opened the segment with translation provided from the memory. Sentences were identified as an 80% match with the difference highlighted in the AnyMem window:



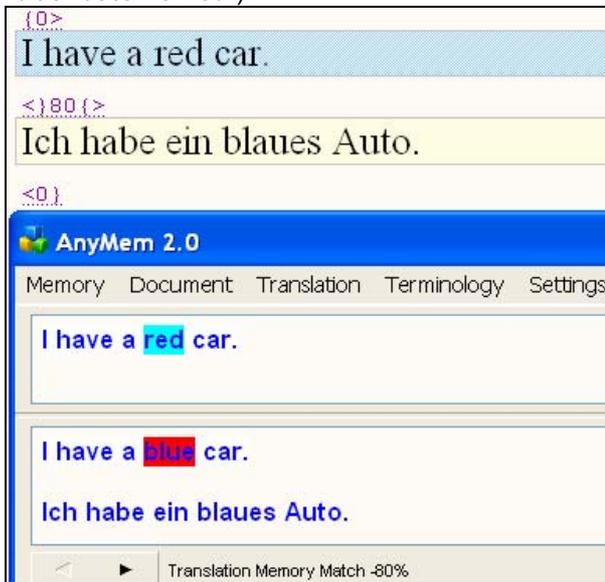
- 6. Thus, instead of translating text all over again we only make additions:



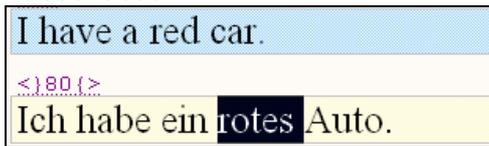
- 7. After translation is edited click the **Next Segment** button:



- 8. The program finds partial equivalent again and highlights the difference between sentences (former "blue" became "red"):



9. Edit translation:

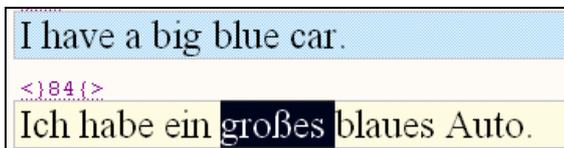


10. Go to the next segment:

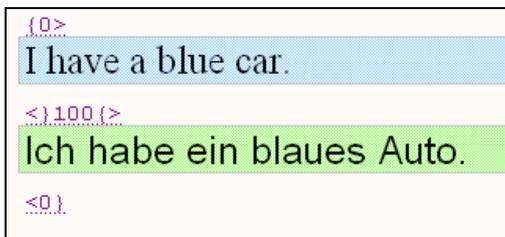


This time concurrence is higher (84%)

11. Again, edit translation:



12. The next segment's background appears green:



This means that AnyMem found a complete match in the memory, and translator has only to make sure that the translation is correct.

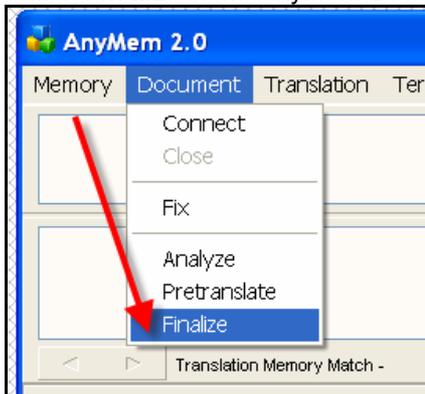
13. If the segment is the last one, the translation of the file is completed:

```
{0>I have a car.<}0{> Ich habe ein Auto.<0}
{0>I have a blue car.<}80{> Ich habe ein blaues Auto.<0}
{0>I have a red car.<}80{> Ich habe ein rotes Auto.<0}
{0>I have a big blue car.<}84{> Ich habe ein großes blaues Auto.<0}
{0>I have a blue car.<}100{> Ich habe ein blaues Auto.<0}
```

Finishing

Before translation is delivered to the client, the file should be cleaned from the source text and segmentation markers to give the document a presentable look. To do so perform the following steps:

1. Save and close the translated document
2. Select **Finalize** in the AnyMem **Document** menu:



3. Add the translated file to the list and click **Finalize**:



4. After the finishing the text in the file will look as shown below:

Ich habe ein Auto.
Ich habe ein blaues Auto.
Ich habe ein rotes Auto.
Ich habe ein großes blaues Auto.
Ich habe ein blaues Auto.

This is a "clean" translated file that can be sent to the client (if a clean file is required).

In fact, that is all you need to start working.

Program operation will be detailed in following sections.

Buttons description

Now we know the principle of program operation let us examine functionality of the toolbar buttons.



(Start translation) – pressing this button will open the first translation segment.



(Next Segment) – transfers to the next segment.



(Previous Segment) – returns to the previous segment.



(Finish Translation) – this button is required when a translation should be finished before the last segment is reached. Pressing this button closes the translated segment and records it to the memory; the next segment is not opened.



(Get From Source) – pressing this button copies the source text into the translation field. The button is especially useful when some text fragment should be left in the source language (e.g., client's address).



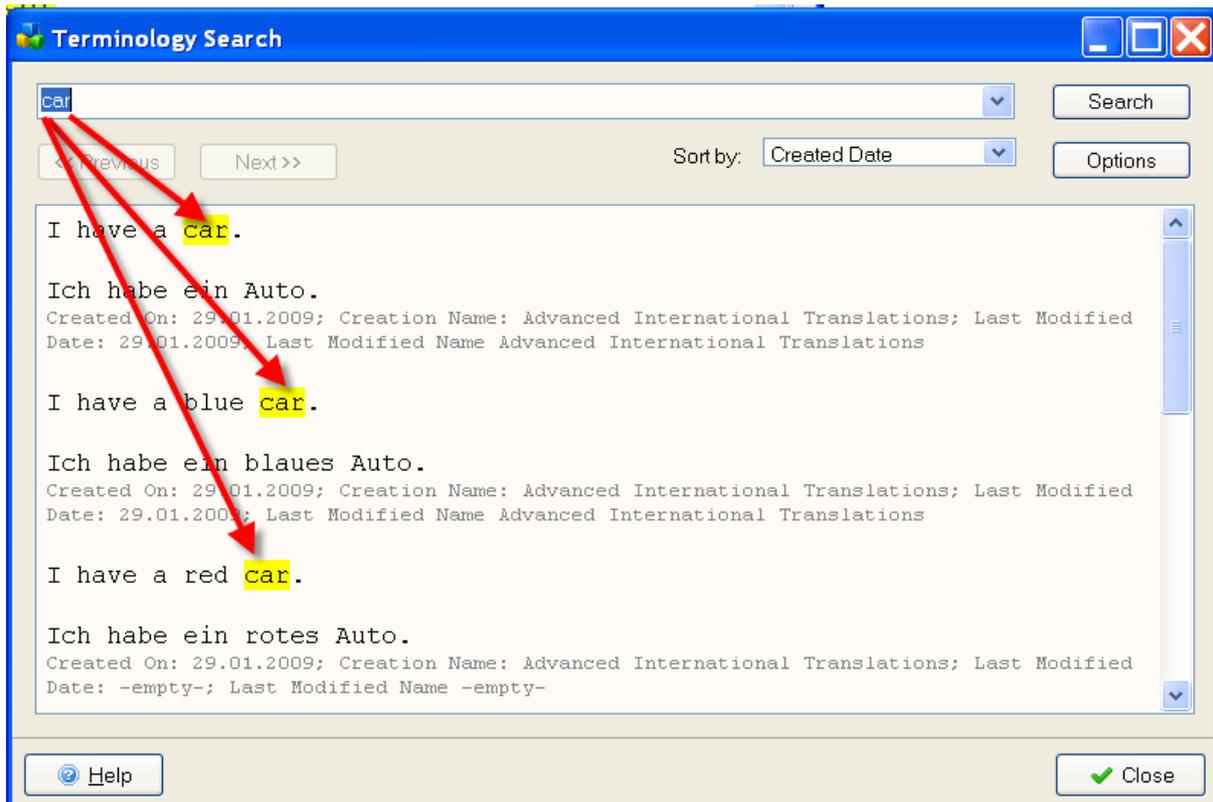
(Get From Memory) – pressing this button enters the text from the memory that corresponds most to the current source into the translation string. This function is useful when the translator does not like own corrections of the previous translation and wants to revert to the translation saved in the memory.



(Interrupt Translation) – upon pressing this button the segmentation of the current segment is deleted and only the source text is left. Thus the text is reverted to the state before translation.



(Terminology Search) – if a word of the source text being translated in the file is selected and then this button is pressed, an additional window appears showing the list of all instances when this word has been translated:



This function is very useful as it saves the translator long and onerous recollection of how a certain term was translated several days ago. And if the memory is passed on to another translator, the latter can quickly find out how the term was translated by the predecessor.

Keyboard shortcuts

Clicking each toolbar button with the mouse is rather time-consuming, therefore we recommend using the following key combinations to save your time:

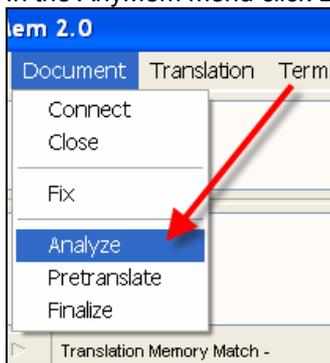
-  (Start translation) – **Alt+Home**.
-  (Next Segment) – **Alt + Num+** (the "+" button on the numeric keypad)
-  (Finish Translation) – **Alt+End**.
-  (Get From Source) – **Alt+Insert**.
-  (Terminology Search) – **Alt + Up Arrow**

Analysis of translation files

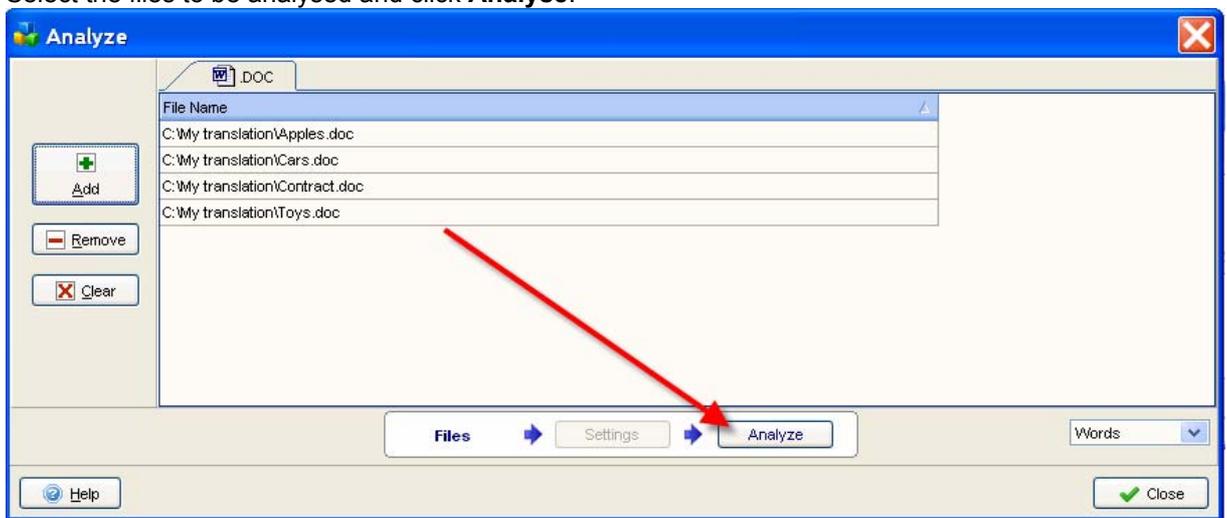
AnyMem allows to analyse the number of words to be translated in a file.

To analyse the volume of work do the following:

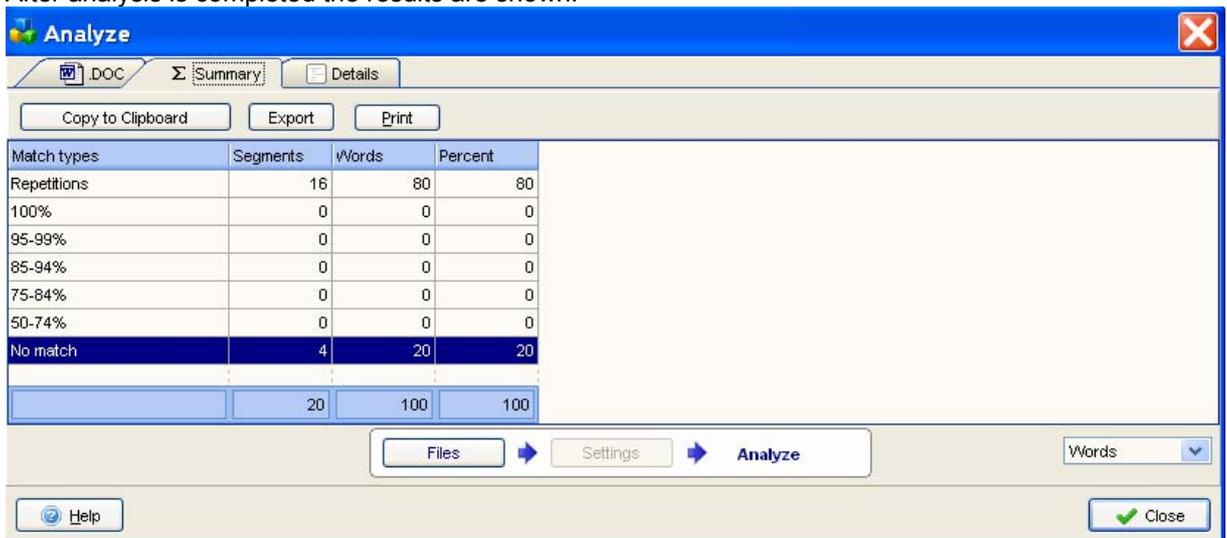
1. In the AnyMem menu click **Document**, and then click **Analyze**:



2. Select the files to be analysed and click **Analyze**:



3. After analysis is completed the results are shown:

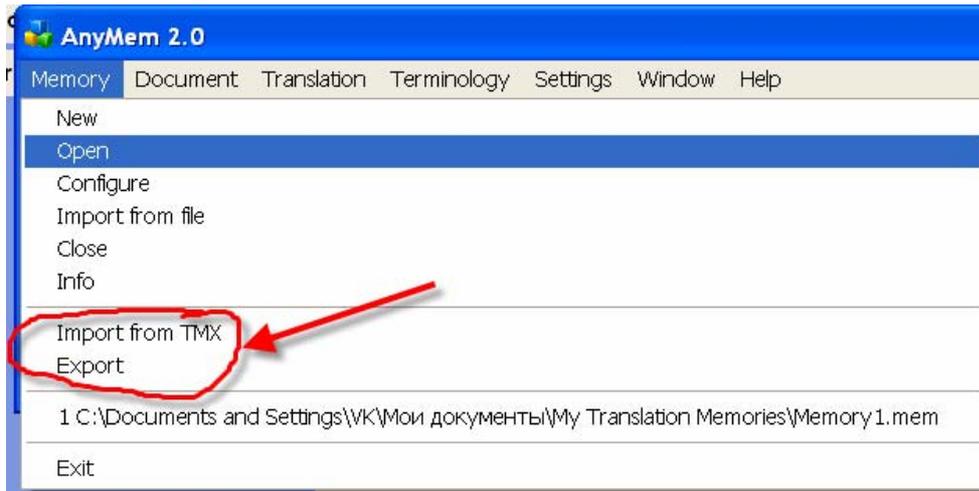


In the example the total number of words in the files is 100, but only 20 words are to be translated with 80 words repeated.

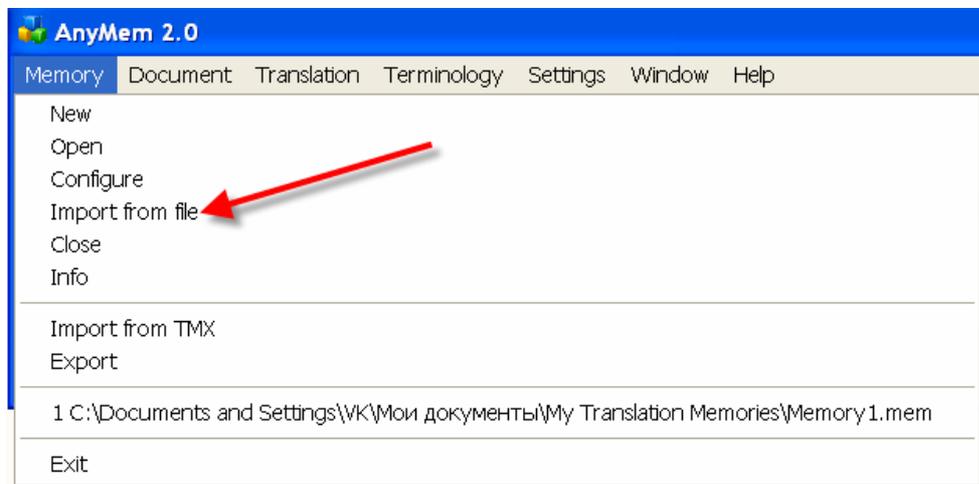
Based on such analysis the translator can estimate the actual volume of work which can be much smaller than it may seem.

Memory export and import

Memory files created during working process can be exported into TMX format, and TMX files can be imported into another translation memory. This option can be especially useful when several translators are engaged in the project and all their work needs to be merged into a single memory file. These actions can be performed with the help of dedicated menus:

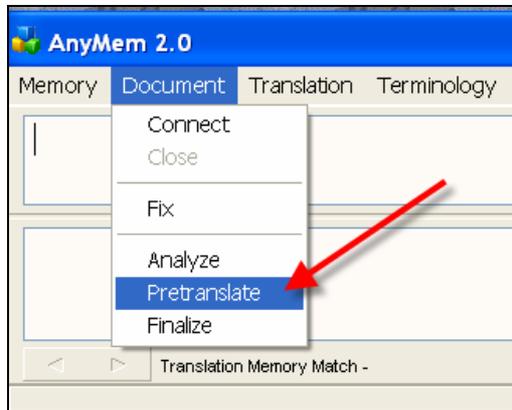


Moreover, a file translated in MS Word can be added to the memory:



Pretranslation

With AnyMem you can automatically translate the text in the files matching with the text in memory.



Summary

Knowledge of the main AnyMem functions described hereinabove should be sufficient to start actively using the program. For further details on functionality see program Help files.