39

Handling dictionaries for FM-12 to FM-16

Overview

FMdict allows to maintain FrameMaker dictionaries both of the old and new format. Some of the functions can also be performed with the standard **Spelling Checker** dialogues, but with more effort.

Contents

Some terminology

|	
Dictionary	See *FrameMaker dictionary environment* on page 15
Language directory	A directory within the dictionary named either all or according to a language-locale code, for example en_GB or de_CH.
Language file	In all language directories you find the file added.txt which contains the learned words. Another file is exceptions.txt.
Project dictionary	This is a concept introduced by **FMdict**. The current dictionary can be saved to a user defined directory.
Site dictionary	The dictionary resulting from reading and interpreting the file specified in maker.ini.
User dictionary	The dictionary into which FM learns words.

Script installation

1 For proper display **FMdict** uses the font `Consolas`. If it does not exist on your system, please install it for better appearance of items in the dialogues.

2 Close FrameMaker if necessary.

3 Download the Inst-**FMdict**.zip from my web-page

4 UnZip the file to the desk top.

5 Execute the file `PrepareInstallation.exe` as Admin.

Note: *Windows Defender may react to this program with a virus warning. This is a false positive. You may check this with an upload to Virustotal. See Avoid Windows Defender intervention on page 3.*

6 The last six FM-versions present in your `%appdata%/Adobe/ FrameMaker/` are listed in the dialogue (on your system there may be only one)

Note: **FMdict** *cooperates only with that FM-installation, that can be reached by double-click of an* `*.fm` *file..*
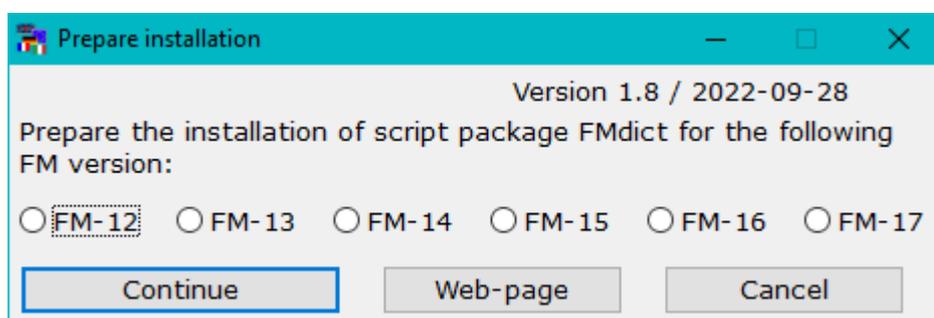
The language of the installation dialogues adapts to the UI language of your current FM installation (de, en, fr).

7 Select the appropriate FM installation and use **Continue** . This opens the installation program:
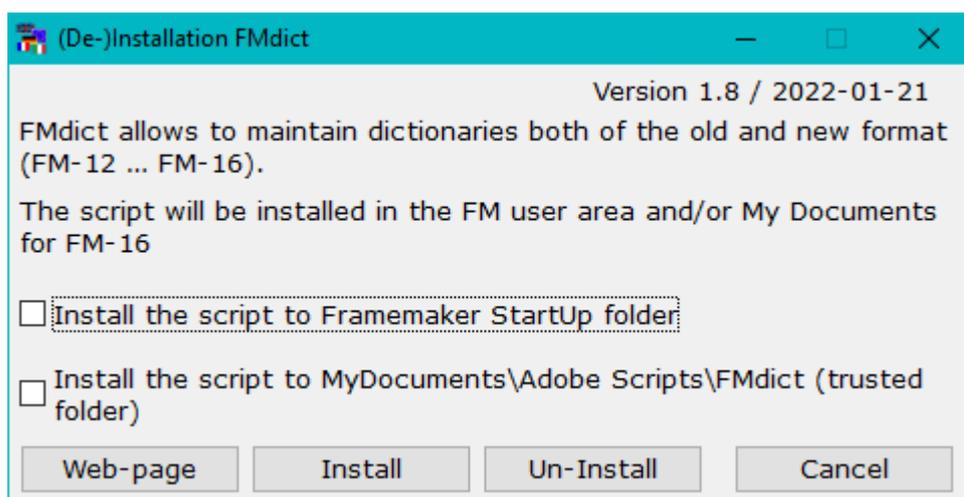
8 Select the appropriate options (See *Inform the FM user about the current dictionary* on page 12).

- Use the first option to start the information-script automatically at FM-start.

- Use the second option if you wish to start the information-script on demand from a trusted source. You can define the script in **File > Scripts ... > Catalog** (Script

Library) as Favourite: use the Add button and navigate to `My Documents \Adobe Scripts\`**FMdict\FMdict.**`jsxbin.`

9 With **Install** the script and associated files will be transferred into the relevant locations.

10 A message indicates the end of the installation. The desktop icons will be removed.

De-install the script

To de-install the script, start the installation program from the short cut in the **Start-menu > D+DD**. Then use the **Un-Install** button.

This removes the script only from the locations defined in the options. You can later use the installation program again.

To get completely rid of the script you need to remove the following in addition to the de-installation:

• The short cut in the start menu

• Directory `%appdata%\`D+DD**FMdict**

Details of the D+DD script installation

• Unpacking creates a program (`Prepinstallation.exe`) and a directory (`InstallThis`).

• `Prepinstallation.exe` should be executed with admin rights (right-click and select), otherwise a message will appear indicating this.

• `Prepinstallation` adds the information about the current directory (e.g. `C:\Users\Klaus\Desktop\InstallThis`) and the installation directory of the selected FM version to the `InstallThis\config.ini` file.

• Then `InstallThis\Inst-`**FMdict.**`exe` is executed, which creates a directory D+DD**FMdict** in the Windows Start directory and copies everything from `InstallThis` there. In addition, a short cut is created to this directory.

• This directory contains `Inst-`**FMdict.**`exe`, which is now executed and the unpacked files that are no longer used (e.g. on the desktop) are deleted.

• `Inst-`**FMdict.**`exe` asks where the script should be copied to (`Startup` folder or `MyDocuments\Adobe Scripts\`**FMdict**). The selected option must also be selected during a de-installation in order to de-install at the correct location.

• The script consists of the main module FMmarkers.jsx and the modules in the **FMdict** directory.

• Various scripts (e.g. the installation of the tool bars) also write something in the `maker.ini` in the user area in the section `[D+DD]`.

Avoid Windows Defender intervention

1 In Windows settings navigate to **Windows Security > Virus & Threat protection settings > Manage Settings**

2 Go to **Exclusions**

3 Add the folder you need to exclude:
`C:\Users\`*username*`\AppData\Roaming\`D+DD

Starting FMdict

To start the utility, use the short-cut `FMdict` in the Windows start-up folder `D+DD`.

Maintaining dictionaries

Use cases

- **FMdict** is especially useful to maintain dictionaries for clients a consultant is working for.
- **FMdict** checks for modification of the User dictionary when it becomes active — hence **FMdict** can work in parallel with FrameMaker.
- The dictionary structure for older FM versions (<11) is different to that of FM-11 and above, hence could not be handled with **FMdict**.

Caveat   At FM start the site dictionary is imported and interpreted. In this process the hyphenation points are recreated and may differ significantly from those in the source file referenced in `maker.ini`. Hence all work done with **FMdict** on the site dictionary is lost with the next start of FM!

Since the resulting internal file does not contain a BOM (Byte Order Mark), feature `Only new words` can not work correctly.

*Note:*   *For background information see FrameMaker dictionary environment on page 15.*

Working with FMdict

The intended process with this utility is:

- Select a source dictionary: User dictionary or Site dictionary1) or a dictionary stored as project [1].
- The dictionary contains various sub-directories for languages and a language-independent one with the name all. You work on one of these subdirectories containing the dictionary words.
- Possible actions on the word list are: Add a new word, Modify a word (replace) and Delete words.
- A selection of words may be moved from the current list (typically the all list) into other subdirectories (typically for a specific language).

---

1   see Caveat on page 1

Main dialogue

```
FMdict [1.7] — Handling FrameMaker dictionaries          —   □   ×

Dictionary type
◉ User          ○ Project  [                    ▼]    ○ Site
C:\Users\Klaus\AppData\LocalLow\Adobe\Linguistics\UserDictionaries…

Language
all      ▲   \FMUserDictionary\all\added.txt        Loaded from project: FM-dict-test
ar_AE
bg_BG
ca_ES
cs_CZ
da_DK        List displays …      ◉ All words        ○ Only new words
de_CH    ▼

Dictionary contents                    Selected / new word
[              ] [\\]                   Words contain at most 63 characters
                                        including the hyphenation symbols.
Fire-Wire™        ▲   [=>]
Frame-Maker                            • Notate hyphenation points with the symbol «-»
Frame-Script                          • Notate the compound hyphen with the symbols «\-»
Frame-Viever                          • Indicate Do-not-hyphenate with starting symbol «-»
Förder-leistung
Führung-schie-nen         [<=]        [  Replace  ]  [   Add   ]  [  Delete  ]
Gesamt-farb-sätti-gung
Hel-veti-ca                            Move selected words to target dictionary
Laser-Writer                           ◉ User dictionary       all    ▲
Minis-ter-posten                                               ar_AE
Poly-line              [=>]           (The site dictionary is loaded at FM-start,  bg_BG    [Move to
WYSI-WYG               [=>]           hence rebuilt at that time - moving there is  ca_ES    dictionary/
au-thor-ing            [=>]           not rational.)          cs_CZ    language]
back-quote                                                     da_DK
car-ry\-over                           ○ Project dictionary    de_CH
coun-ter-in-tu-i-tive                                          de_DE    Done!
de-mo-ing        ▼                    [                    ▼]  el_GR
                                                               en_CA ▼

[  Save as project  ]           [ Help (pdf) ]  [ Anc. functions ]  [  Close  ]
```

- As long as you have not distributed words into the various language sub-directories only category all contains words.
- Button **Help (pdf)** opens this document at the appropriate page.
- In the dialogue the hyphenation points are visualised same as in the FM dialogues (- and \-).
- If the dictionary file is empty the word **E M P T Y** will be the only one in the list. You can remove such empty dictionary folders with **Anc. Functions > Current Project**.

Select the source dictionary

- Depending on the source the actions differ:

Source is User or Site	Source is Project dictionary [a]
Select either User or Site dictionary.	Select Project . ➜ The dropdown list becomes active
	Select the desired project from the drop down list
	➜ 'Loaded from project" is visible.
Select the desired language. The list of languages may be short by removing all _empty' language directoreis.(see *Ancillary functions* on page 9	

a. See *Project dictionary* on page 12 for the definition of such a dictionary.

Selecting the language file

After defining the type of dictionary you need to define from which subdirectory - defined by language - the words shall be taken.

All words or only new ones?

ll words are already displayed in the word list. You may, however, select Only new words to get a shorter list. The word list was never been saved the list indicated E M T P Y.

Selecting in the word list

> *Note:* *The word list is saved automatically if the dialogue becomes inactive (for example, when you switch to FrameMaker). The word list is updated automatically when the dialogue becomes active again.*

A special mechanism is necessary to allow for two different list handling actions:

- Find words in a long list by typing a few characters
- Select multiple words.

Find word

- In the edit field above the list type the first few characters (case independent) and terminate them with a blank.
- Be aware that the list starts with the words not to be hyphenated, which is indicated by a preceding minus.
- Either the word is found and selected or the first word in the list is selected.
- To start a new search you first need to clear the search by using the **** button. It is not sufficient to just clear theinput field.

Multiple selection

To select more than one word in the list you can use the standard Windows technique with **SHIFT+arrow**, **CTRL+click** and/or **SHIFT+CTRL+click**.

- Multiple words are only reasonable for moving them to a target dictionary.
- Only the first word in a selection is transferred to the edit field.

Modify a word in the list

- Select a word in the list
- Use button **=>** to copy the word into the editing area.
- Modify the word. For example change hyphenation or correct spelling.
- Use button **Replace** to move it back into the list.
- The maximum length of the word including hyphenation is checked.

Add a new word to the list

- Type the new word into the editing area - or get a 'base' word from the list via button **=>**.
- If the word shall not be hyphenated, precede it with a hyphen.
- The hyphen for compound words is preceded by a backslash (\–).
- Use button **Add** to move the word into the list.
- The maximum length of the word including hyphenation is checked [2].

Remove words from the list

- You can use this function only for the complete list (**All words**).
- Select a word or group of words (use **SHIFT+CTRL+click**).

---

2 `Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz` can be place only without distinct hyphenation points (63 characters). But this one is to long: `Donaudampfschifahrtselektrizitätenhauptbetriebs–werkbauunterbeamtengesellschaft` (79).

- Use the **Delete** button to remove the selected words from the list.

Move to a target dictionary/ language

- In the list select the word or words to be moved.
- Depending on the target the actions differ:

Source is User	Target is Project dictionary [a]
To the right of the three => symbols select **User** dictionary.	To the right of the three => symbols select **Project** dictionary ➜ The drop down list becomes active
Select the desired language. This list contains all possible languages. ➜ Button **Move** becomes active.	
User button **Move to dictionary/language**	
The word(s) will be removed from the source dictionary and appended to the target dictionary. If the target dictionary does not exist it is created in new format.	

a. See *Project dictionary* on page 12 for the definition of such a dictionary.

Save modified word list

The following actions will *automatically* save a modified word list:

- Switching between dictionary type (User, ...)
- Switching between the languages in a dictionary.
- Switching the word list from **All words** to **Only new words**.
- Switching task to another application (e.g. FrameMaker).
- Quitting the dialogue with **Close** (or the **X** button).

Leaving the dialogue with Close or [X]

All necessary saves have already been done, hence the dialogue will just close.

Save As Project

Save as Project in the main panel opens a panel with two tabs: This panel allows to save the current dictionary structure to a project structure.

Save to existing

This saves the current dictionary structure to a given project-folder.

Note: *This overwrites the target structure! Hence you will normally use this function only after loading from that project.*
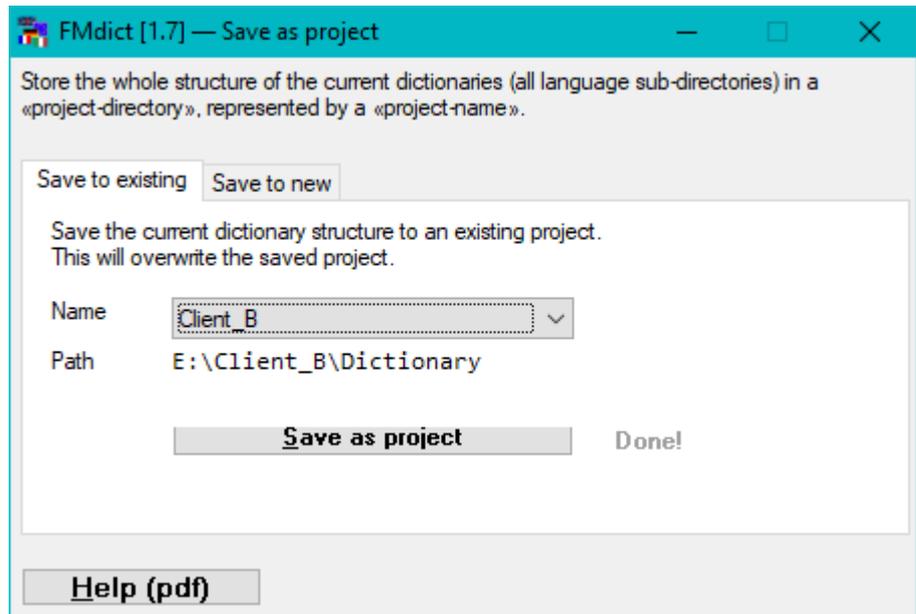
Name	Select an existing project from the drop down list.
Path	The corresponding directory path will be displayed.
Save as project	Perform the action.

Save to new

This saves the current dictionary structure to a new project folder.
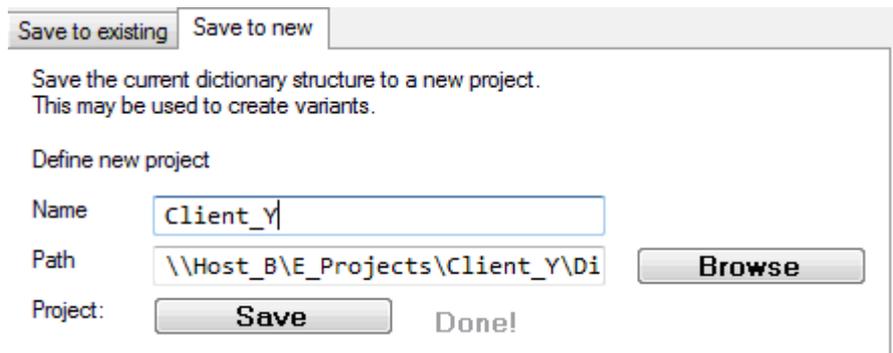
Name	Provide a name for reference to this project structure.
Path	Either enter the path directly or use the Browse button to find a proper location.
Save	The file structure is copied from the current dictionary to the defined path.

Ancillary functions

Button **Anc. Functions** in the main panel opens a panel with 6 tabs:
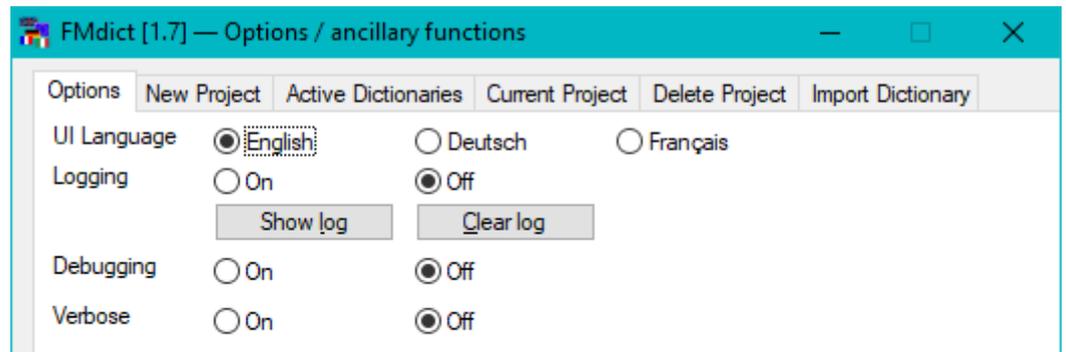
Options

UI Language	This may have been set already by the installation routine.
Logging	If On, certain actions are logged in a file:

- Start of the utility.
- Adding, deleting or replacing words in the word list.
- Loading dictionaries from a Project dictionaries.
- Saving the current word list.

Note: *The settings of the radio buttons become active only after the next start of the utility.*

Show log	The current log file FM-dict-log.txt is opened in the standard text editor. Clear the log with **Clear log**, not by deleting the file.
Clear log	This clears the contents and then writes en entry with the date of this action:

`2017-09-22 15:53:00 Log cleared`

Debugging	Set to **on** only by advice of the author in case of problems.
Verbose	Debug output also from within loops and lowest level functions. This also sets **Debugging** to ON.

New Project

Set up a Project dictionary in advance. This is a file structure for all possible languages. (see *Relevant file structure* on page 17). The files are empty.
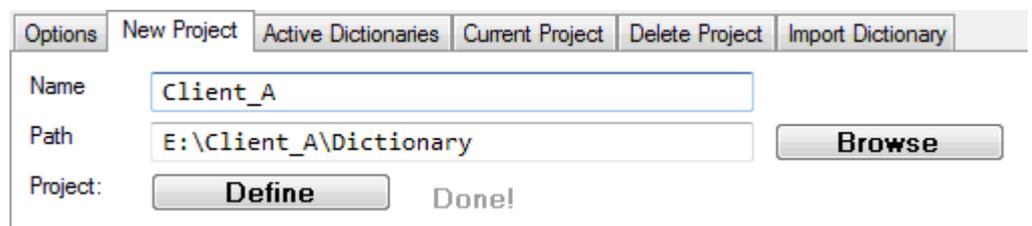
Name	Provide a name for reference to this project.
Path	Either enter the path directly or use the **Browse** button to find a proper location.
Define	With this button the file structure is copied from the script-library to the defined path.

Active dictionaries

Active dictionaries are the file structures in FMUserDictionary and FMSiteDictionary. Most language directories in there

contain empty language files. These useless language directories can be removed here.
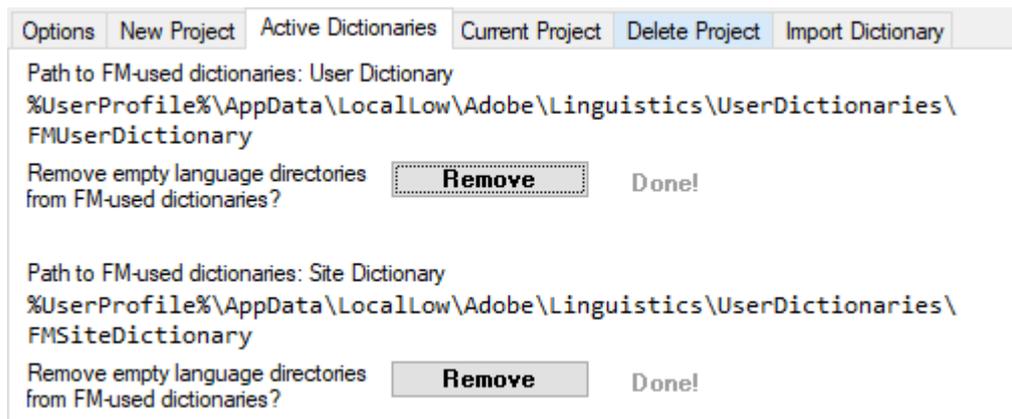
Current Project

You can remove the not-used language sub-directories of the current project. This provides better overview in the language list after loading into the User dictionary.
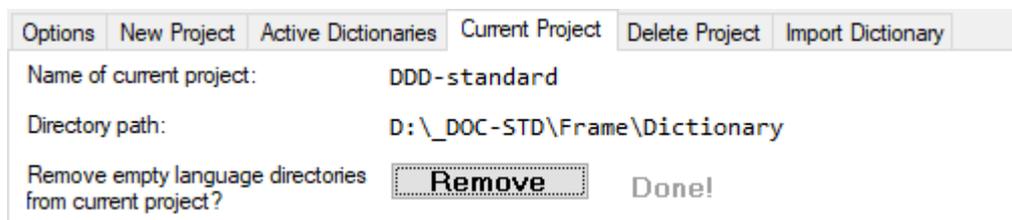
Delete Project

After finishing work for a client you may wish to remove the project from the list and optionally remove all files (backup!)
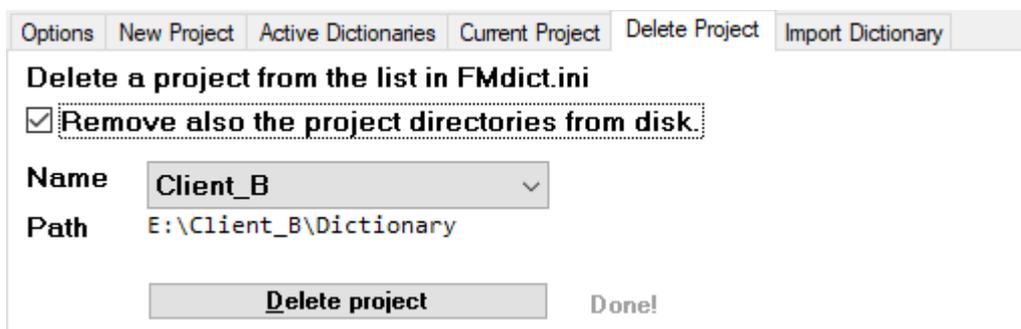
Import word lists

Service providers and consultants often need to handle data from older FrameMaker versions. Therefore they may have FM-dictionary files saved in old format.

Another case to use this function is the collection of Unknown words with the **Spelling checker > Dictionaries > Write all Unknown Words to File...**[3]

These input files are of different formats which are automatically handled by the routine thanks to the header lines:

Unknown words   `<MakerDictionary 3.0>`

Old dictionary   `<MakerDictionary 2.0>`

Internal files   no header, `\x02` user for hyphen

---

3   3These words are unknown to the whole dictionary environment (Provide dictionaries, User, Site, document) - all the 'red wiggled' ones.

With this mechanism it is also possible to read the site dictionary (`sitedict.dct`) and later move its items into a language of the user dictionary or the all pseudo-language.

| Options | New Project | Active Dictionaries | Current Project | Delete Project | Import Dictionary |

Import a word list file into the pseudo-language aa_imp in the current user dictionary.

Path   `C-STD\Frame\Dictionaries-v2\ddd-lorem.dct`   **Browse**

Project: **Import**   Done!
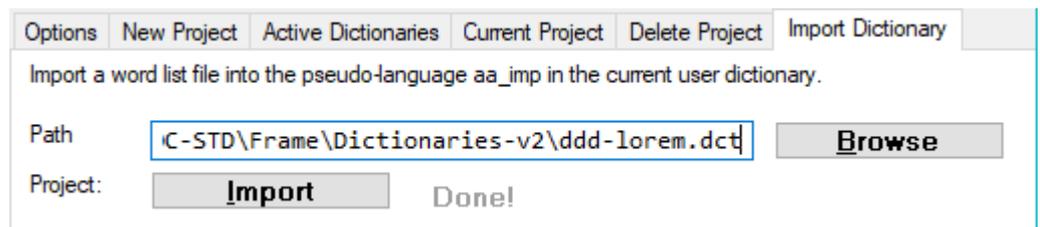
The words are transferred into the pseudo-language `aa_imp` of the User dictionary. If you are already in the User dictionary you will see the new 'language' after selecting **User dictionary** again. Import speed is about 1000 words/second (2.5 GHz).

You can then, for example, save the current user dictionary to a (new) **FMdict** project.

Handle new languages

The combination of Write all unknown words to File, Import word lists and Save/load **FMdict** project allows to support new languages, for example Latin:

1 Take a document using only the language in question (e.g classic Latin). Define the paragraph language to something far away of the language in question. E.g.to get the Latin words all unknown set the paragraph language to Russian.

2 To get a large word list you may copy into the document as many source files you can get hold of (e.g. from the web).

3 Use **Spelling checker > Dictionaries > Write all Unknown Words to File…**

4 In a text editor you may see all words with starting hyphen - they will not be hyphenated. I have removed the initial hyphens.

5 In **FMdict** perform these steps:
   - Set up a new project, let's say: `MyLatin`.
   - Restart the **FMdict** to see the new project in the list.
   - Load this project. All languages are empty.
   - Import the word list from step 2.
   - Switch to User dictionary.
   - You will get the pseudo-language aa_imp.
   - Move all these words to User dictionary, language all.
   - Save your project (`MyLatin`). Your User dictionary now contains all the Latin words from the text corpus without any hyphenation points.

6 If You come along a document with the prepared language, just load the appropriate **FMdict** project and continue to add words with FrameMaker or **FMdict**.

Arbitrary word lists

Yo may have word lists, for example to train Liang's hyphenation algorithm in T$_E$X. It is easy to convert them with a text editor to the new dictionary form (see *Source dictionary* on page 13) and import them.

Project dictionary

A project dictionary is a storage location for the personal dictionary (`FMUserDictionary`) used for a particular client or project.

Use case

Assuming that you are currently working for **Client_A**. You have now finished that work or need to work for another client. Hence you want to save the dictionary accumulated in the current project. Store it away as a project dictionary.

The utility supports the following structure of a project dictionary (which is the same structure as the User dictionary):

Structure	Example
Project-directory\	E:\Client_A\dictionary\
All	All\
added.txt	added.txt
Language_1	en_GB\
added.txt	added.txt
Language_x	de_DE<
added.txt	added.txt
...	...

With button **Save Project** you save the current personal dictionary (all language subdirectories) to the project dictionary. If the target does not yet exist, you are prompted to define a directory and a name of the project. This data is kept in `FM-dict.ini`.

To work for **Client_Q** you load that project dictionary (copy the file structure to the `FMUserDictionary` folder on which FrameMaker works.

Current project

You could ruin the current `FMUserDictionary` by loading another project dictionary. To avoid this, the current dictionary project is noted in `FMdict.ini`, which is checked.

```
[FM-dict]
currentPrj =Client_A
currentDir =E:\Client_A\Dictionary
```

Creating a new FMdict project

You can create an **FMdict** project in advance - before you really store something in it. You do this with **Anc. Functions > New Project**.

Save current User dictionary as FMdict project

The current dictionary may be from any of the sources given in the top row of radio buttons. Using **Save as project** saves the complete user dictionary (`FMUserDictionary`) to a specified directory.

Inform the FM user about the current dictionary

A start-up script informs the user about source of the User dictionary. Menu **File > Dictionary information** does the same.
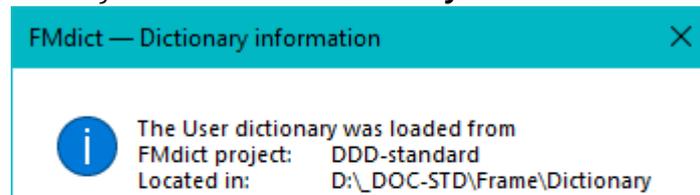
This requires a function available only in FM greater v11.

Program details

This utility must be something outside FM (hence no Extend-Script), because otherwise FM updates the files at its closing. **FMdict** is implemented as an AHK script. It is distributed in the compiled form.

- The dialogue can not be resized[4]. And it assumes the existence of the font `Consolas`.
- The dialogue starts in the user dictionary.

Source dictionary
- A loaded dictionary can be of old or new format. At the close of the dialogue or Save it will be saved always in new format.
- Reading dictionary files accounts for three formats:

	Old format	New format	
Header line	`<MakerDictionary 2.0>`	`<MakerDictionary 3.0>`	- none -
Coding	Windows 1252 using Frame Roman	UTF-8	UTF-8
Hyphenation	-	-	\x02
Composita	\-	\-	-
Usage	`site.dct` in $HOME	Exported dictionaries or "unknown words".	Internal dictionaries (in the user area)

- The dictionary files may be original with UNIX line-end (`\x0A`) or touched already by Windows (line-end is `\x0D0A`). Reading the dictionary into the word-list accounts for this.
- It's quite code intensive to get the location of the site dictionary. Not only that the registry must be consulted - we must consider two locations of `maker.ini`.
- FM has no problem to handle a new format file where initially was an old format file (site dictionary).
- The file resulting from importing the site dictionary does not carry a `BOM`. This hinders the compare routine `FunGetDiffFile` to create a correct short list. There is no way in any programming language to detect whether a UTF file carries `BOM` or not! FM bug ticket is open.

Word list
The **=>** button transfers only the first word from a selection to the editing area.

- A selection of words can be transferred to another dictionary/language category.
- The sort order in the word list is the same as in FM dictionaries:
  - The dictionary file start with the lines of words not to be hyphenated,
  - Then follow words with upper case initial.
  - Words with lower case initials follow.
  - Last words use non-Latin script
- The list of new words has no corresponding file in which the item could be deleted.
- While a `ListBox` provides multiple selection, a `ComboBox` provides the incremental search. Since we need to be able to

---

4   This would be quite tedious programming. Think about how long it took Adobe to make at least some dialogues resizeable …

find words in a long list by typing a few characters and be able to select multiple words, a special mechanism is nec- essary. It is based on the work of AHK guru PhiLho.

Saving the word list

Before the current word list is saved, the original is renamed to `filename_saved.extension`. In case of the site dictionary[5] saving may lead to a conversion to the new format.

Working in parallel with FrameMaker

You may have **FMdict** open during your work with Frame-Maker.

- FrameMaker updates the User Dictionary[6] whenever you learn a word or define special hyphenation for a word.
- When leaving the dialogue of **FMdict** (the window becomes inactive) to work with FrameMaker (or another application), **FMdict** automatically saves the current word list.
- When you come back from FrameMaker to FM-dict (the window becomes active) **FMdict** checks the time stamp of the User dictionary to see whether it had been modified. If so, the word list is refreshed. **Only new words** will then display the words newly entered by FrameMaker.

Messages

Closing the message window will normally return control back to the dialogue.

MsgDelCurrent

This action will delete the current project!

MsgDictChanged

The User Dictionary has been modified since loading from project «*ProjName*» ...

MsgHaveLoaded

The current User Dictionary has been loaded from a project «*ProjName*» To keep the current content: Cancel. ...

MsgNewCurrent

The current dictionary was loaded from project «*ProjName1*». Do you want to save it as another project «*ProjName2*» ...

MsgNoFile

This file does not exist: «*filename*». Try another one.

MsgNonTxtFile

The file extension of the target file «*filename*» is not `txt` - program will not write to file!.

MsgNoOldFile

No saved version of this dictionary exist. ...

MsgNothingSelected

Nothing has been selected, hence nothing can be moved.

MsgPathInvalid

Given path «*ProjPath*» is invalid - no drive!

MsgPathUsed

Sorry project location already used for «*ProjPath*». Choose another location or project name.

MsgSameFile

The target file can not be the same as the source file «*filename*». ...

MsgSaveAsProj

Store the whole structure of the current dictionaries (all language sub-directories) in a «project-directory», represented by a «project-name».

MsgSelProj

The current dictionary was loaded from project «*ProjName*» and is unchanged. ...

MsgWordLimit

The word «*word*» is longer (*wLen*) than allowed: *maxWordLen* characters including hyphenation points).

---

5 Not tested!
6 Currently only the file `\all\added.txt` is updated. So this is checked.

FrameMaker dictionary environment

These sections provide background information.

Dictionary types

Note: *The dictionary types are listed here in the order of look-up.*

Document dictionary

The document dictionary is a part in the document, which can be seen in MIF (since FM-8 MIF files are UTF-8):

```
<Dictionary
   <OKWord `Blakeslee'>
   <OKWord `bodytext'>
   <OKWord `bulleting'>
   <OKWord `Ctrl+Alt+Shift+click'>
   <OKWord `HOME\\fminit\\maker'>
   <OKWord `Positioniergenauigkeit'>
   <OKWord `Steinfressergrünling'>
   <OKWord `ungrouped'>
> # end of Dictionary
```

You see that there are no hyphenation points defined and languages are mixed …

My findings show that the words for the document dictionary are maintained (*allowed in document*, unlearn) in the file `C:\Users\User\AppData\LocalLow\Adobe\Linguistics\User Dict ionaries\FMDocDictionary\all\added.txt` and integrated into the document file (`fm` or `mif`) at the time of saving.

User dictionary

Learned words are accumulated in one UTF-8 file, independent of language:
`C:\Users\User\AppData\LocalLow\Adobe\Linguistics\User Dict ionaries\FMUserDictionary\all\added.txt`

Words in this file *do not contain hyphenation points*, even if there are optional hyphens in the word in the document which is learnt.

```
Positionierungsgenauigkeit
```

It is possible to define a user dictionary in the `maker.ini` file. I must assume that it creates the same problems as the site dictionary.

Site dictionary

The site dictionary can not be handled by the Spelling Checker dialogue. Its location is defined in the `maker.ini` file:

```
SiteDictionary=dict\site.dct
```

Relative paths are relative to `$HOME`. Absolute paths can also point to the network.

After installation of FM the file is in the old format (see *Import a personal (user) dictionary* on page 18). It can, however, also be in new format. In the new format even Russian words can be included:

```
здоровье                (health)
```

At the start of FrameMaker the site dictionary is (if old format: converted and) imported to

`C:\Users\User\AppData\`**`LocalLow`**`\Adobe\Linguistics\`**`User Dictionaries`**`\FMSiteDictionary\all\added.txt`. This replaces the previous content.

Note: *The transfer, however, does not take over hyphenation points. This is independent of format (old, new, with/out header). They are set anew and may create very strange results:*

```
Ge¦s¦a¦mt¦far¦bsätti¦gung
```

Note: *This behaviour makes it impossible to maintain the site diction-ary! ➜ Bug FRMAKER-3201.*
The problem can not be overcome by setting the entry in `maker.ini` *to nothing. The internal file will be cleared in this case.*

Provider dictionaries

For most languages two providers are available: `Proximity` and `Hunspell`. The files with the dictionaries and the hyphena-tion information are located in (starting wit FM-12)
`$HOME\fminit\Linguistics\7.0\Providers`

These are consulted if a word is not found in one of the FM-maintained dictionaries (document, user, site).

Project dictionaries

This concept is introduced by **FMdict**. It is intended for service providers dealing with various clients. They must maintain cli-ent-specific dictionaries. Currently this is done by manually copying the user-dictionaries around. See *Project dictionary* on page 12 for more details.

Relevant file structure

At the installation of FM the following files are installed and/ or updated.

%UserProfile%\...\Linguistics

`C:\Users\userName\AppData\LocalLow\Adobe\Linguistics\`

This is the location of the FrameMaker dictionaries - not those of the 'dictionary providers'. The structure is nearly identical to that in
`C:\Users\userName\AppData\Roaming\Adobe\Linguistics\` 7)

In the lowest levels many language subdirectories are left out here (indicated by an ellipsis):

```
├──Dictionaries
│   ├──Adobe Custom Dictionary
│   │   ├──all
│   │   ...
│   │   └──sgr
│   ├──FMDocDictionary
│   │   ├──all
│   │   ...
│   │   └──sgr
│   ├──FMSiteDictionary
│   │   ├──all
│   │   ...
│   │   └──sgr
│   └──FMUserDictionary
│       ├──all
│       ...
│       └──sgr
└──UserDictionaries
    ├──Adobe Custom Dictionary
    │   ├──all
    │   ├──ar_AE
    │   ...
    │   ├──tr_TR
    │   └──uk_UA
    ├──FMDocDictionary
    │   ├──all
    │   ├──ar_AE
    │   ...
    │   ├──tr_TR
    │   └──uk_UA
    ├──FMSiteDictionary
    │   ├──all
    │   ├──ar_AE
    │   ...
    │   ├──tr_TR
    │   └──uk_UA
    └──FMUserDictionary
        ├──all
        ├──ar_AE
        ...
        ├──tr_TR
        └──uk_UA
```

---
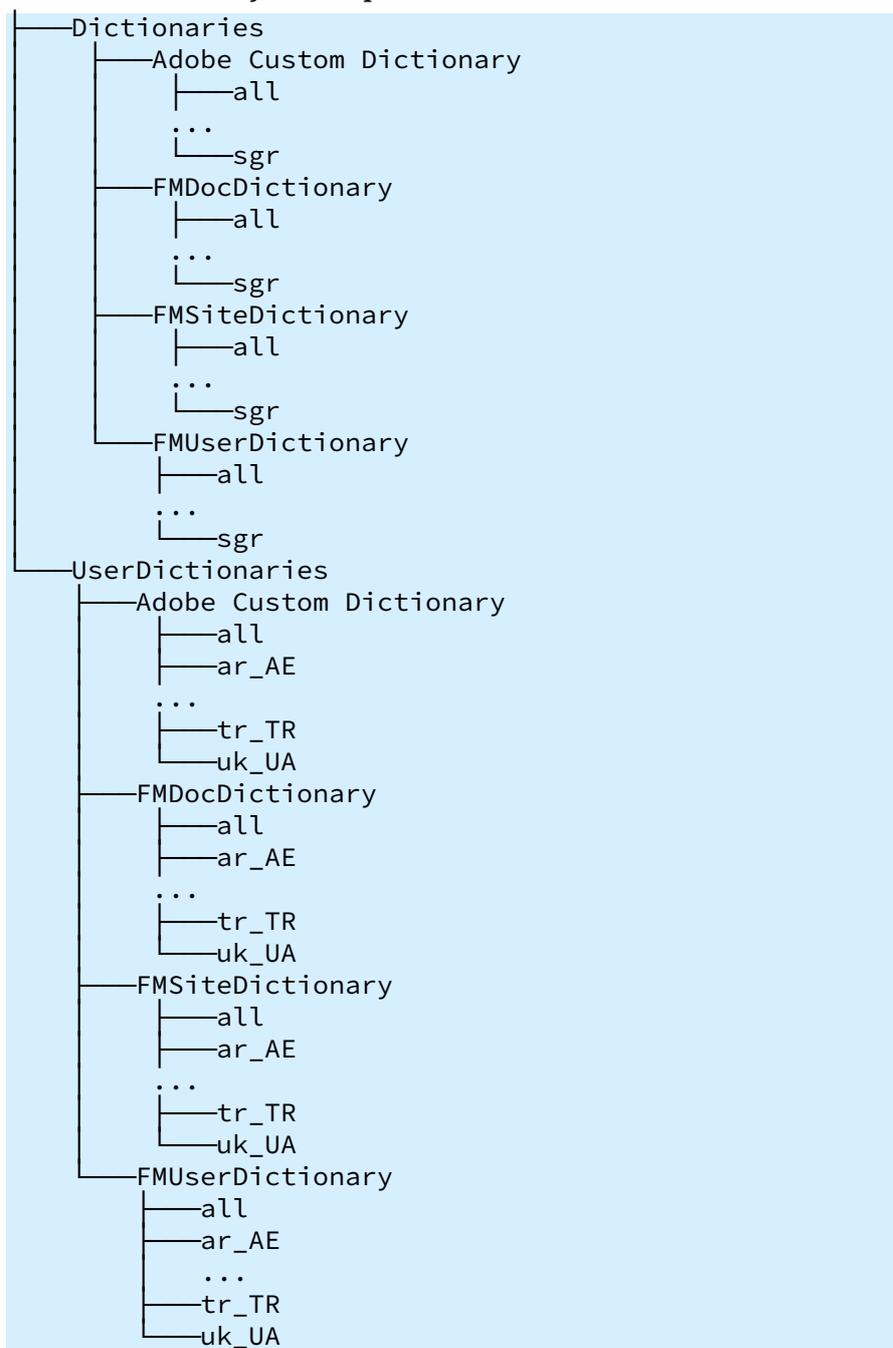
7   The purpose of this structure is unknown. It can safely be renamed in FM-12 upwards. Maybe it is used by older FM versions.

E:_DDDprojects\FM-Dict\Docu\FMdict-en.fm

Dictionary operations

Writing user dictionary to file

Spelling Checker > Dictionaries… > Personal Dictionary write to File … lets you export the user dictionary to a text file. This will contain hyphenation points:

```
Po-si-tion-i-erungs-genau-ig-keit
```

It's unknown to me, what defines the hyphenation points - which are often disappointing.

The written file is Windows code page 1252 only. For Unicode requiring words it must be converted before it can be used.

Import a personal (user) dictionary

- Importing replaces the whole file added.txt! You use Spelling Checker > Dictionaries… > Personal Dictionary Import dictionary. The dictionary to be imported must be either in the old or in the new format and have these prop- erties:

Old format	New format
Coding Windows 1252	Coding UTF-8
First line:	First line:
`<MakerDictionary 2.0>`	`<MakerDictionary 3.0>`
Hyphenation character is -	Hyphenation character is \x02 [a]
Coupling characters are \-	Coupling character is -
Characters with a code above \x7F must use the Frame Roman coding	All characters are Unicode
`Farb\xd2\x9fber\xd3-s\x8attigung`	`Farb"über"-sättigung`
`-FireWire\xaa`	`¦FireWire™`

a. In this paper the character \x02 is displayed as broken vertical (¦)

- There is no error message if anything goes wrong with the import, for example by wrong coding. The dictionary may be empty and the old contents gone …

Merging a user dictionary

The user dictionary added.txt can receive additional input from another dictionary file via **Spelling Checker > Dictionaries… > Personal Dictionary: Merge from File**.

Example

```
<MakerDictionary 3.0>
Cheybyshev
-Codec
Extend-Script
Kirchen-pflege
Posi-tion-ierung
```

Resulting added.txt

```
¦Codec
¦FireWire™
¦GHz
Chey¦by¦shev
Extend¦Script
Frame¦Script
Förder¦leis¦tung
Führung¦schienen
Gesamt¦farb¦sät¦ti¦gung
Kirchen¦pflege
Posi¦tion¦ierung
Un¦boot¦able
wild¦card
```

2022-09-28

E:_DDDprojects\FM-Dict\Docu\FMdict-en.fm